

The L^AT_EX 2 _{ε} Sources

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

2024-11-01

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

Contents

| | | |
|-----------|---|-----------|
| 01 | ltdirchk.dtx | 1 |
| 1 | L^AT_EX System Dependent Initializations | 1 |
| 2 | Initialization | 2 |
| 2.1 | INITEX | 3 |
| 2.2 | Some bits of 2e | 4 |
| 3 | texsys.cfg | 5 |
| 3.1 | texsys.cfg | 5 |
| 3.2 | UNIX (web2c) | 6 |
| 3.3 | UNIX (other) | 7 |
| 3.4 | MSDOS (emtex) | 7 |
| 3.5 | MSDOS (other) | 7 |
| 3.6 | VMS (DECUS T _E X, PD VMS 3.6) | 7 |
| 3.7 | VMS (???) | 7 |
| 3.8 | MACINTOSH (OzTeX 1.6) | 8 |
| 3.9 | MACINTOSH (other) | 8 |
| 3.10 | FAKE EXAMPLE | 8 |
| 4 | Setting \@currdir | 9 |
| 5 | Setting \input@path | 10 |

| | | |
|-----------|--|-----------|
| 6 | Filename Parsing | 11 |
| 7 | T_EX Versions | 13 |
| 8 | ltxcheck.tex | 13 |
| | | |
| 02 | ltplain.dtx | 14 |
| 1 | Plain T_EX | 14 |
| | | |
| 03 | ltvers.dtx | 37 |
| 1 | Version Identification | 37 |
| 1.1 | Declaring an all-new module | 40 |
| | | |
| 04 | ltluatex.dtx | 42 |
| 1 | Overview | 42 |
| 2 | Core T_EX functionality | 42 |
| 3 | Plain T_EX interface | 43 |
| 4 | Lua functionality | 43 |
| 4.1 | Allocators in Lua | 43 |
| 4.2 | Lua access to T _E X register numbers | 44 |
| 4.3 | Module utilities | 45 |
| 4.4 | Callback management | 45 |
| 5 | Implementation | 46 |
| 5.1 | Minimum LuaT _E X version | 46 |
| 5.2 | Older L ^A T _E X/Plain T _E X setup | 47 |
| 5.2.1 | Fixes to <i>etex.src/etex.sty</i> | 47 |
| 5.2.2 | luatex specific settings | 48 |
| 5.3 | Attributes | 49 |
| 5.4 | Category code tables | 49 |
| 5.5 | Named Lua functions | 51 |
| 5.6 | Custom whatsis | 51 |
| 5.7 | Lua bytecode registers | 52 |
| 5.8 | Lua chunk registers | 52 |
| 5.9 | Lua loader | 52 |
| 5.10 | Lua module preliminaries | 54 |
| 5.11 | Lua module utilities | 54 |
| 5.11.1 | Module tracking | 54 |
| 5.11.2 | Module messages | 55 |
| 5.12 | Accessing register numbers from Lua | 56 |
| 5.13 | Attribute allocation | 57 |
| 5.14 | Custom whatsit allocation | 58 |

| | | |
|-----------|--|------------|
| 5.15 | Bytecode register allocation | 58 |
| 5.16 | Lua chunk name allocation | 58 |
| 5.17 | Lua function allocation | 59 |
| 5.18 | Lua callback management | 59 |
| 5.18.1 | Housekeeping | 59 |
| 5.18.2 | Handlers | 64 |
| 5.18.3 | Public functions for callback management | 67 |
| 05 | ltexpl.dtx | 73 |
| 1 | expl3-dependent code | 73 |
| 1.1 | Loader | 73 |
| 1.2 | Using expl3 code | 76 |
| 2 | Document-level command names for expl3 functions | 77 |
| 06 | ltdefns.dtx | 80 |
| 1 | Definitions | 80 |
| 1.1 | Initex initializations | 80 |
| 1.2 | Saved versions of T _E X primitives | 80 |
| 1.3 | Command definitions | 81 |
| 1.4 | Robust commands and protect | 90 |
| 1.5 | Acting on robust commands | 96 |
| 1.5.1 | Copying robust commands | 98 |
| 1.5.2 | Showing robust commands | 100 |
| 1.5.3 | Commands defined with \DeclareRobustCommand | 101 |
| 1.5.4 | Commands defined with \newcommand (with optional argument) | 103 |
| 1.5.5 | Showing environments | 104 |
| 1.6 | Internal defining commands | 106 |
| 2 | Discretionary Hyphenation | 110 |
| 07 | ltcmd.dtx | 113 |
| 1 | Creating document commands | 113 |
| 1.1 | Variables and constants | 113 |
| 1.2 | Declaring commands and environments | 117 |
| 1.3 | Structure of xparse commands | 122 |
| 1.4 | Normalizing the argument specifications | 127 |
| 1.5 | Preparing the signature: general mechanism | 135 |
| 1.6 | Setting up a standard signature | 136 |
| 1.7 | Setting up expandable types | 141 |
| 1.7.1 | Copying a command and its internal structure | 144 |
| 1.7.2 | Showing the definition of a command | 150 |
| 1.8 | Grabbing arguments | 156 |
| 1.9 | Grabbing arguments expandably | 168 |
| 1.10 | Argument processors | 174 |

| | | |
|-----------|--|------------|
| 1.11 | Conversion to key–value form | 176 |
| 1.12 | Utilities | 179 |
| 1.13 | Messages | 183 |
| 1.14 | User functions | 188 |
| 08 | lthooks.dtx | 194 |
| 1 | Introduction | 194 |
| 2 | Package writer interface | 194 |
| 2.1 | $\text{\LaTeX} 2\epsilon$ interfaces | 194 |
| 2.1.1 | Declaring hooks | 194 |
| 2.1.2 | Special declarations for generic hooks | 195 |
| 2.1.3 | Using hooks in code | 196 |
| 2.1.4 | Updating code for hooks | 197 |
| 2.1.5 | Hook names and default labels | 199 |
| 2.1.6 | The <code>top-level</code> label | 202 |
| 2.1.7 | Defining relations between hook code | 202 |
| 2.1.8 | Querying hooks | 204 |
| 2.1.9 | Displaying hook code | 205 |
| 2.1.10 | Debugging hook code | 206 |
| 2.2 | L3 programming layer (<code>exp13</code>) interfaces | 206 |
| 2.3 | On the order of hook code execution | 209 |
| 2.4 | The use of “reversed” hooks | 211 |
| 2.5 | Difference between “normal” and “one-time” hooks | 212 |
| 2.6 | Generic hooks provided by packages | 212 |
| 2.7 | Hooks with arguments | 213 |
| 2.8 | Private \LaTeX kernel hooks | 215 |
| 2.9 | Legacy $\text{\LaTeX} 2\epsilon$ interfaces | 215 |
| 3 | $\text{\LaTeX} 2\epsilon$ commands and environments augmented by hooks | 216 |
| 3.1 | Generic hooks | 216 |
| 3.1.1 | Generic hooks for all environments | 217 |
| 3.1.2 | Generic hooks for commands | 218 |
| 3.1.3 | Generic hooks provided by file loading operations | 218 |
| 3.2 | Hooks provided by <code>\begin{document}</code> | 219 |
| 3.3 | Hooks provided by <code>\end{document}</code> | 219 |
| 3.4 | Hooks provided by <code>\shipout</code> operations | 221 |
| 3.5 | Hooks provided for paragraphs | 221 |
| 3.6 | Hooks provided in NFSS commands | 221 |
| 3.7 | Hook provided by the mark mechanism | 222 |

| | | |
|-----------|---|------------|
| 4 | The Implementation | 222 |
| 4.1 | Debugging | 222 |
| 4.2 | Borrowing from internals of other kernel modules | 223 |
| 4.3 | Declarations | 223 |
| 4.4 | Providing new hooks | 225 |
| 4.4.1 | The data structures of a hook | 225 |
| 4.4.2 | On the existence of hooks | 226 |
| 4.4.3 | Setting hooks up | 227 |
| 4.4.4 | Disabling and providing hooks | 233 |
| 4.5 | Parsing a label | 235 |
| 4.6 | Adding or removing hook code | 239 |
| 4.7 | Setting rules for hooks code | 260 |
| 4.8 | Specifying code for next invocation | 281 |
| 4.9 | Using the hook | 283 |
| 4.10 | Querying a hook | 289 |
| 4.11 | Messages | 292 |
| 4.12 | L ^T E _X 2 _E package interface commands | 295 |
| 4.13 | Deprecated that needs cleanup at some point | 299 |
| 4.14 | Internal commands needed elsewhere | 301 |
| 09 | ltcmdhooks.dtx | 303 |
| 1 | Introduction | 303 |
| 2 | Restrictions and Operational details | 304 |
| 2.1 | Patching | 305 |
| 2.1.1 | Timing | 305 |
| 2.2 | Commands that look ahead | 305 |
| 3 | Package Author Interface | 306 |
| 3.1 | Arguments and redefining commands | 307 |
| 4 | The Implementation | 307 |
| 4.1 | Execution plan | 307 |
| 4.2 | Variables | 308 |
| 4.3 | Variants | 309 |
| 4.4 | Patching or delaying | 309 |
| 4.5 | Patching commands | 311 |
| 4.5.1 | Patching by expansion and redefinition | 312 |
| 4.5.2 | Patching by retokenization | 320 |
| 4.6 | Messages | 326 |
| 10 | ltsockets.dtx | 328 |
| 1 | Introduction | 328 |

| | |
|---|------------|
| 2 Configuration of the transformation process | 328 |
| 2.1 The template mechanism | 328 |
| 2.2 The hook mechanism | 329 |
| 2.3 The socket mechanism | 329 |
| 2.3.1 Examples | 331 |
| 2.3.2 Details and semantics | 332 |
| 2.3.3 Command syntax | 334 |
| 2.3.4 Rationale for error handling | 335 |
| 3 The Implementation | 336 |
| 3.1 Debugging the socket structures | 336 |
| 3.2 The L3 layer commands | 337 |
| 3.3 Error messages | 340 |
| 3.4 The L ^T E _X 2 _ε interface commands | 340 |
| 11 lttemplates.dtx | 342 |
| 1 Introduction | 342 |
| 2 What is a document? | 342 |
| 3 Types, templates, and instances | 343 |
| 4 Template types | 343 |
| 5 Templates | 343 |
| 6 Multiple choices | 346 |
| 7 Instances | 347 |
| 8 Document interface | 348 |
| 9 Changing existing definitions | 348 |
| 10 Ad hoc adjustment of templates | 349 |
| 11 Getting information about templates and instances | 349 |
| 12 The implementation | 350 |
| 12.1 Variables and constants | 351 |
| 12.2 Testing existence and validity | 353 |
| 12.3 Saving and recovering property lists | 354 |
| 12.4 Creating new template types | 356 |
| 12.5 Design part of template declaration | 357 |
| 12.5.1 Storing values | 360 |
| 12.6 Implementation part of template declaration | 361 |
| 12.7 Editing template defaults | 366 |
| 12.8 Creating instances of templates | 368 |
| 12.9 Using templates directly | 370 |
| 12.10 Assigning values to variables | 371 |

| | | |
|-----------|---|------------|
| 12.11 | Using instances | 374 |
| 12.12 | Assignment manipulation | 374 |
| 12.13 | Showing templates and instances | 375 |
| 12.14 | Messages | 376 |
| 12.15 | User functions | 380 |
| 12 | ltalloc.dtx | 382 |
| 1 | Counters | 382 |
| 13 | ltcntrl.dtx | 384 |
| 1 | Program control structure | 384 |
| 14 | lterror.dtx | 388 |
| 1 | Error handling and tracing | 388 |
| 1.1 | General commands | 388 |
| 1.2 | Specific errors | 394 |
| 1.3 | Tracing | 398 |
| 15 | ltpar.dtx | 399 |
| 1 | Paragraphs | 399 |
| 1.1 | Implementation | 399 |
| 16 | ltpara.dtx | 401 |
| 1 | Introduction | 401 |
| 1.1 | The default processing done by the engine | 401 |
| 2 | The new mechanism implemented for L ^A T _E X | 403 |
| 2.1 | The provided hooks | 404 |
| 2.2 | Altered and newly provided commands | 405 |
| 2.3 | Examples | 406 |
| 2.3.1 | Testing the mechanism | 406 |
| 2.3.2 | Mark the first paragraph of each <code>itemize</code> | 408 |
| 2.4 | Some technical notes | 408 |
| 2.4.1 | Glue items between paragraphs (found with <code>fancypar</code>) | 408 |
| 3 | The Implementation | 409 |
| 3.1 | Providing hooks for paragraphs | 409 |
| 3.2 | The error messages | 416 |
| 17 | ltmeta.dtx | 418 |

| | | |
|-----------|---|------------|
| 1 | Introduction | 418 |
| 1.1 | \DocumentMetadata | 418 |
| 2 | The Implementation | 418 |
| | | |
| 18 | ltspace.dtx | 421 |
| 1 | Spacing | 421 |
| 1.1 | User Commands | 421 |
| 1.2 | Chris' comments | 421 |
| 1.3 | Some immediate actions | 423 |
| 1.4 | The code | 424 |
| 1.5 | Vertical spacing | 431 |
| 1.6 | Horizontal space (and breaks) | 437 |
| | | |
| 19 | ltlogos.dtx | 441 |
| 1 | Logos | 441 |
| | | |
| 20 | ltfiles.dtx | 442 |
| 1 | File Handling | 442 |
| 1.1 | Safe Input Macros | 455 |
| 1.2 | Listing files | 463 |
| | | |
| 21 | ltoutenc.dtx | 466 |
| 1 | Font encodings | 466 |
| 1.1 | Removing encoding-specific commands | 468 |
| 1.2 | The order of declarations | 469 |
| 1.3 | Docstrip modules | 469 |
| 1.4 | Definitions for the kernel | 470 |
| 1.4.1 | Declaration commands | 470 |
| 1.4.2 | Hyphenation | 478 |
| 1.4.3 | Miscellania | 478 |
| 1.4.4 | Default encodings | 478 |
| 1.4.5 | Math material | 481 |
| 1.5 | Definitions for the OT1 encoding | 482 |
| 1.6 | Definitions for the T1 encoding | 484 |
| 1.7 | Definitions for the OMS encoding | 490 |
| 1.8 | Definitions for the OML encoding | 491 |
| 1.9 | Definitions for the OT4 encoding | 491 |
| 1.10 | Definitions for the TS1 encoding | 493 |
| 1.11 | Definitions for the TU encoding | 498 |
| | | |
| 2 | Package files | 509 |
| 2.1 | The fontenc package | 509 |

| | | |
|-----------|--|------------|
| 22 | ltcounts.dtx | 512 |
| 1 | Counters and Lengths | 512 |
| 1.1 | Environment Counter Macros | 512 |
| 23 | ltlength.dtx | 521 |
| 1 | Lengths | 521 |
| 24 | ltfssbas.dtx | 523 |
| 1 | Preliminary macros | 523 |
| 2 | Macros for setting up the tables | 524 |
| 3 | Selecting a new font | 531 |
| 3.1 | Macros for the user | 531 |
| 3.2 | Macros for loading fonts | 536 |
| 4 | Assigning math fonts to <i>versions</i> | 544 |
| 25 | ltfssaxes.dtx | 551 |
| 1 | Changing the font series | 551 |
| 1.1 | The series lookup table | 551 |
| 1.2 | Mapping rules for series changes | 552 |
| 1.3 | Changing to a new series | 560 |
| 2 | Changing the shape | 565 |
| 2.1 | Mapping rules for shape combinations | 566 |
| 2.2 | Changing to a new shape | 568 |
| 3 | Make sure we win ... | 570 |
| 26 | ltfsstrc.dtx | 572 |
| 1 | Introduction | 572 |
| 2 | A driver for this document | 572 |
| 3 | The Implementation | 573 |
| 4 | Handling Options | 573 |

| | | |
|-----------|--|------------|
| 5 | Macros common to <code>fam.tex</code> and <code>tracefnt.sty</code> | 575 |
| 5.1 | General font loading | 575 |
| 5.2 | Math fonts setup | 581 |
| 5.2.1 | Outline of algorithm for math font sizes | 581 |
| 5.2.2 | Code for math font size setting | 582 |
| 5.2.3 | Other code for math | 583 |
| 6 | Scaled font extraction | 585 |
| 6.1 | Sizefunctions | 593 |
| 27 | ltfsscmp.dtx | 597 |
| 28 | ltfssdcl.dtx | 602 |
| 1 | Interface Commands | 602 |
| 29 | ltfssini.dtx | 634 |
| 1 | NFSS Initialization | 634 |
| 1.1 | Providing math <i>versions</i> | 634 |
| 2 | Custom series settings for main document families | 635 |
| 3 | Supporting nested emphasis | 652 |
| 3.1 | Legacy | 656 |
| 3.2 | Miscellaneous | 656 |
| 30 | fontdef.dtx | 662 |
| 1 | Introduction | 662 |
| 2 | Customization | 662 |
| 3 | The <code>docstrip</code> modules | 663 |
| 4 | A driver for this document | 663 |
| 5 | The <code>fonttext.ltx</code> file | 663 |
| 5.1 | Encodings | 664 |
| 5.2 | Defaults | 666 |

| | | |
|-----------|--|------------|
| 6 | The <code>fontmath.ltx</code> file | 668 |
| 6.1 | The font encodings used | 668 |
| 6.1.1 | Symbolfont and Alphabet declarations | 669 |
| 6.2 | Math font sizes | 669 |
| 6.3 | The math symbol assignments | 670 |
| 6.3.1 | The letters | 670 |
| 6.3.2 | The digits | 671 |
| 6.3.3 | Punctuation, brace, etc. keys | 671 |
| 6.3.4 | Delimitercodes for characters | 672 |
| 6.4 | Symbols accessed via control sequences | 672 |
| 6.4.1 | Greek letters | 672 |
| 6.4.2 | Ordinary symbols | 673 |
| 6.4.3 | Large Operators | 674 |
| 6.4.4 | Binary symbols | 674 |
| 6.4.5 | Relations | 675 |
| 6.4.6 | Arrows | 676 |
| 6.4.7 | Punctuation symbols | 677 |
| 6.4.8 | Math accents | 678 |
| 6.4.9 | Radicals | 678 |
| 6.4.10 | Over and under something, etc | 678 |
| 6.4.11 | Delimiters | 679 |
| 6.5 | Math versions of text commands | 680 |
| 6.6 | Other special functions and parameters | 680 |
| 6.6.1 | Biggggg | 680 |
| 6.6.2 | The log-like functions | 681 |
| 6.6.3 | Parameters | 681 |
| 7 | Default cfg files | 681 |
| 31 | preload.dtx | 683 |
| 1 | Overview | 683 |
| 2 | Customization | 683 |
| 3 | Module switches for the <code>DOCSTRIP</code> program | 683 |
| 4 | A driver for this document | 684 |
| 5 | The code | 684 |
| 32 | ltfntcmd.dtx | 686 |
| 1 | Introduction | 686 |
| 2 | The implementation | 688 |
| 3 | Initialization | 694 |

| | | |
|-----------|---|------------|
| 33 | lttextcomp.dtx | 695 |
| 1 | Sub-encodings | 699 |
| 1.1 | Unavailable in sub-encoding 1 and higher (drop symbols not working in Latin Modern) | 700 |
| 1.2 | Unavailable in sub-encoding 2 (majority of new OTF fonts via autoinst) and higher | 700 |
| 1.3 | Unavailable in sub-encoding 3 and higher | 704 |
| 1.4 | Unavailable in sub-encoding 4 and higher | 704 |
| 1.5 | Unavailable in sub-encoding 5 (most older PS fonts) and higher | 704 |
| 1.6 | Unavailable in sub-encoding 6 and higher | 705 |
| 1.7 | Unavailable in sub-encoding 7 and higher | 705 |
| 1.8 | Unavailable in sub-encoding 8 and higher | 705 |
| 1.9 | Unavailable in Sub-encoding 9 (most missing) | 705 |
| 2 | Unicode engine specials | 706 |
| 3 | Font family sub-encodings setup | 706 |
| 4 | Legacy symbol support for lists and footnote symbols | 711 |
| 5 | The <code>textcomp</code> package | 716 |
| 5.1 | The old <code>textcomp</code> package code | 717 |
| 5.1.1 | Supporting oldstyle digits | 725 |
| 5.1.2 | Subset encoding defaults | 725 |
| 6 | The <code>checkencodingsubset.tex</code> file | 727 |
| 34 | ltpageno.dtx | 737 |
| 1 | Page Numbering | 737 |
| 35 | ltxref.dtx | 738 |
| 1 | Cross Referencing | 738 |
| 1.1 | Cross Referencing | 738 |
| 36 | ltproperties.dtx | 746 |
| 1 | Introduction | 746 |
| 2 | Design discussion | 746 |
| 3 | Handling unknown labels and properties | 747 |
| 4 | Rerun messages | 747 |
| 5 | Open points | 747 |

| | | |
|-----------|---|------------|
| 6 | Code interfaces | 747 |
| 7 | Auxiliary file interfaces | 749 |
| 8 | L^AT_EX 2_≤ interface | 749 |
| 9 | Pre-declared properties | 750 |
| 10 | The Implementation | 751 |
| 10.1 | Reference commands | 753 |
| 10.2 | Tests and warnings | 755 |
| 10.3 | Predeclared properties | 757 |
| 10.4 | Messages | 759 |
| 37 | ltmisen.dtx | 760 |
| 1 | Miscellaneous Environments | 760 |
| 1.1 | Environments | 760 |
| 1.2 | Center, Flushright, Flushleft | 774 |
| 1.3 | Verbatim | 777 |
| 38 | ltmath.dtx | 785 |
| 1 | Math setup | 785 |
| 1.1 | Math commands based on plain T _E X | 785 |
| 1.1.1 | The log-like functions | 785 |
| 1.1.2 | Biggggg | 786 |
| 1.1.3 | The UNSORTED Rest | 786 |
| 1.2 | Math Environments | 793 |
| 1.3 | External options to the standard document classes | 798 |
| 1.3.1 | Left equation numbering | 798 |
| 1.3.2 | Flush left equations | 798 |
| 39 | ltlists.dtx | 801 |
| 1 | List, and related environments | 801 |
| 1.1 | List and Trivlist | 802 |
| 1.2 | Vertical Spacing (skips) | 803 |
| 1.3 | Penalties | 803 |
| 1.4 | Horizontal Spacing (dimens) | 803 |
| 1.5 | Default Values | 803 |
| 1.6 | Itemize and Enumerate | 816 |
| 40 | ltboxes.dtx | 818 |
| 1 | L^AT_EX Box commands | 818 |
| 1.1 | Some low-level constructs | 836 |

| | | |
|-----------|--|------------|
| 41 | lttab.dtx | 837 |
| 1 | Tabbing, Tabular and Array Environments | 837 |
| 1.1 | tabbing | 837 |
| 1.2 | array and tabular environments | 846 |
| 42 | ltpictur.dtx | 862 |
| 1 | Picture Mode | 862 |
| 1.1 | Curves | 889 |
| 43 | ltthm.dtx | 894 |
| 1 | Theorem Environments | 894 |
| 44 | ltsect.dtx | 898 |
| 1 | Sectioning Commands | 898 |
| 1.1 | The Title | 898 |
| 1.2 | Sectioning | 899 |
| 1.2.1 | Initializations | 906 |
| 1.3 | Table of Contents etc. | 906 |
| 1.3.1 | Convention | 906 |
| 1.3.2 | Commands | 906 |
| 45 | ltfloat.dtx | 911 |
| 1 | Floats | 911 |
| 1.1 | Floating Environments | 911 |
| 1.2 | Footnotes | 925 |
| 46 | ltidxglo.dtx | 934 |
| 1 | Index and Glossary Generation | 934 |
| 47 | ltbibl.dtx | 937 |
| 1 | Bibliography Generation | 937 |
| 1.1 | Default definitions | 941 |
| 48 | ltmarks.dtx | 942 |
| 1 | Introduction | 942 |

| | | |
|-----------|--|------------|
| 2 | Design-level and code-level interfaces | 943 |
| 2.1 | Use cases for conditionals | 945 |
| 2.2 | Debugging mark code | 946 |
| 3 | Application examples | 946 |
| 4 | Legacy L^AT_EX 2_≤ interface | 946 |
| 4.1 | Legacy design-level and document-level interfaces | 946 |
| 4.2 | Legacy interface extensions | 947 |
| 5 | Notes on the mechanism | 947 |
| 6 | Internal output routine functions | 949 |
| 7 | The Implementation | 950 |
| 7.1 | Allocating new mark classes | 950 |
| 7.2 | Updating mark structures | 952 |
| 7.3 | Placing and retrieving marks | 959 |
| 7.4 | Comparing mark values | 961 |
| 7.5 | Messages | 962 |
| 7.6 | Debugging the mark structures | 962 |
| 7.7 | Designer-level interfaces | 964 |
| 8 | L^AT_EX 2_≤ integration | 965 |
| 8.1 | Core L ^A T _E X 2 _≤ integration | 965 |
| 8.2 | Other L ^A T _E X 2 _≤ output routines | 968 |
| 8.3 | Rollback information | 968 |
| 49 | ltpage.dtx | 969 |
| 1 | Page styles and related commands | 969 |
| 1.1 | Page Style Commands | 969 |
| 1.2 | How a page style makes running heads and feet | 969 |
| 1.3 | marking conventions | 969 |
| 50 | ltclass.dtx | 974 |
| 1 | Introduction | 974 |
| 2 | User interface | 974 |
| 2.1 | Option processing | 975 |
| 3 | Class and Package interface | 976 |
| 3.1 | Class name and version | 976 |
| 3.2 | Package name and version | 976 |
| 3.3 | Requiring other packages | 976 |
| 3.4 | Declaring new options | 977 |
| 3.5 | Safe Input Macros | 978 |

| | | |
|-----------|--|-------------|
| 4 | Implementation | 978 |
| 4.1 | Hooks | 1007 |
| 4.2 | Providing shipment | 1010 |
| 5 | Package/class rollback mechanism | 1018 |
| 6 | After Preamble | 1026 |
| 51 | ltkeys.dtx | 1027 |
| 1 | Creating and using keyval options | 1027 |
| 1.1 | Implementation of <code>ltkeys</code> | 1028 |
| 1.2 | Key properties | 1028 |
| 1.3 | Main mechanism | 1029 |
| 1.4 | The document interfaces | 1033 |
| 1.5 | Option usage scope | 1034 |
| 1.6 | General key setting | 1035 |
| 52 | ltfilehook.dtx | 1036 |
| 1 | Introduction | 1036 |
| 1.1 | Provided hooks | 1036 |
| 1.2 | General hooks for file reading | 1036 |
| 1.3 | Hooks for package and class files | 1037 |
| 1.4 | Hooks for <code>\include</code> files | 1038 |
| 1.5 | High-level interfaces for L ^A T _E X | 1039 |
| 1.6 | Kernel, class, and package interfaces for L ^A T _E X | 1040 |
| 1.7 | A sample package for structuring the log output | 1040 |
| 2 | The Implementation | 1041 |
| 2.1 | Document and package-level commands | 1041 |
| 2.2 | <code>expl3</code> helpers | 1042 |
| 2.3 | Declaring the file-related hooks | 1045 |
| 2.4 | Patching L ^A T _E X's <code>\InputIfFileExists</code> command | 1045 |
| 2.5 | Declaring a file substitution | 1047 |
| 2.6 | Selecting a file (<code>\set@curr@file</code>) | 1049 |
| 2.7 | Replacing a file and detecting loops | 1052 |
| 2.7.1 | The Tortoise and Hare algorithm | 1053 |
| 2.8 | Preventing a package from loading | 1055 |
| 2.9 | High-level interfaces for L ^A T _E X | 1056 |
| 2.10 | Internal commands needed elsewhere | 1056 |
| 3 | A sample package for structuring the log output | 1057 |
| 4 | Package emulations | 1058 |
| 4.1 | Package <code>atveryend</code> emulation | 1058 |
| 53 | ltshipout.dtx | 1059 |

| | | |
|-----------|--|-------------|
| 1 | Introduction | 1059 |
| 1.1 | Overloading the <code>\shipout</code> primitive | 1059 |
| 1.2 | Provided hooks | 1060 |
| 1.3 | Legacy L ^A T _E X commands | 1062 |
| 1.4 | Special commands for use inside the hooks | 1063 |
| 1.5 | Provided LuaT _E X callbacks | 1063 |
| 1.6 | Information counters | 1064 |
| 1.7 | Debugging shipout code | 1064 |
| 2 | Emulating commands from other packages | 1065 |
| 2.1 | Emulating <code>atbegshi</code> | 1065 |
| 2.2 | Emulating <code>everyshi</code> | 1066 |
| 2.3 | Emulating <code>atenddvi</code> | 1066 |
| 2.4 | Emulating <code>everypage</code> | 1066 |
| 3 | The Implementation | 1067 |
| 3.1 | Debugging | 1067 |
| 3.2 | Handling the end of job hook | 1079 |
| 4 | Legacy L^AT_EX 2_E interfaces | 1082 |
| 5 | Internal commands needed elsewhere | 1083 |
| 6 | Package emulation for compatibility | 1084 |
| 6.1 | Package <code>atenddvi</code> emulation | 1084 |
| 6.2 | Package <code>atbegshi</code> emulation | 1085 |
| 6.3 | Package <code>everyshi</code> emulation | 1086 |
| 54 | ltoutput.dtx | 1087 |
| 1 | Output Routine | 1087 |
| 1.1 | Floats | 1087 |
| 1.1.1 | Kludgeins | 1143 |
| 1.1.2 | Float control | 1145 |
| 1.1.3 | Float placement parameters | 1158 |
| 55 | lttagging.dtx | 1161 |
| 1 | General support for tagged output | 1161 |
| 2 | Implementation | 1162 |
| 2.1 | Tagging sockets | 1163 |
| 2.1.1 | Tagging support for paragraph setup | 1163 |
| 2.1.2 | Tagging socket for targets | 1164 |
| 2.1.3 | Tagging sockets for toc | 1164 |
| 2.1.4 | Tagging support for table/tabular packages | 1164 |
| 2.1.5 | Tagging Support for floats | 1166 |

| | | |
|-----------------------|---|-------------|
| 3 | For ltab.dtx parked here for now | 1166 |
| 3.1 | Variables for row, column and span counting | 1166 |
| 3.2 | Tracing/debugging | 1167 |
| 3.3 | Interface commands | 1167 |
| 56 | lthyphen.dtx | 1172 |
| 57 | ltfinal.dtx | 1174 |
| 1 | Final settings | 1174 |
| 1.1 | Debugging | 1174 |
| 1.2 | Typesetting parameters | 1174 |
| 1.3 | Lccodes for hyphenation | 1178 |
| 1.4 | Hyphenation | 1180 |
| 1.5 | Font loading | 1181 |
| 1.6 | Input encoding | 1182 |
| 1.7 | Lccodes and uccodes | 1187 |
| 1.8 | Applying Patch files | 1190 |
| 1.9 | Freeing Memory | 1191 |
| 1.10 | Initialise file list | 1191 |
| 1.11 | Preparation for supporting PDF in backends | 1192 |
| 1.12 | Do some temporary work for pre-release | 1192 |
| 1.13 | Some last minute initializations | 1192 |
| 1.14 | Dumping the format | 1192 |
| Change History | | 1193 |
| Index | | 1271 |

File 01

ltdirchk.dtx

1 L^AT_EX System Dependent Initializations

This file implements the semi-automatic determination of various system dependent parts of the initialization. The actual definitions may be placed in a file `texsys.cfg`. Thus for operating systems for which the tests here do not result in acceptable settings, a ‘hand written’ `texsys.cfg` may be produced.

The macros that must be defined are:

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

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

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

So either `{dir}` should be defined as a macro with argument delimited by space, or it should just expand to a directory name, including the final directory separator, so that it may be concatenated with the `{filename}`. This means that for UNIX-like syntax, each `{dir}` should end with a slash, /.

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

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

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

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

`\@TeXversion` `\@TeXversion` is now set automatically by the initialization tests in this file. You should not need to set it in `texsys.cfg`, however the following documentation is left for information. L^AT_EX does not set this variable exactly, the automatic tests set it to:
2 for any version, v , $v < 3.0$
3 for any version, v , $3.0 \leq v \leq 3.14$

`<undefined>` otherwise.

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

If your T_EX is older than version 3.141, then you should define `\@TeXversion` (using `\def`) to be the version number. If you do not do this¹, L^AT_EX will not work around a bug in old T_EX versions, and so error messages will appear in a very strange format, with `^^J` appearing instead of line breaks:

```
LaTeX Error: \rubbish undefined.^^J^^JSee the LaTeX manual or LaTeX=
Companion
for explanation.^^JType H <return> for immediate help.
...
.3 \renewcommand{\rubbish}
{}
```

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

```
LaTeX Error: \rubbish undefined.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
.
...
.3 \renewcommand{\rubbish}
{}
```

Note that this has an extra line `! .` which does not appear in error messages that use the default settings with a current version of T_EX, but this should not cause any confusion we hope.

2 Initialization

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.

Most such definitions are repeated later in the “right” place, usually (but not always) with different implementations. To be able to spot this more easily if you look into the file `latex.ltx` (which is stripped of comments) we add some comment lines to that effect that survive the stripping process by `docstrip`.

```
1 <*dircheck>
2 %% ---- START temporary definitions for bootstrapping; later overwritten ----
3 </dircheck>
```

¹Actually if your T_EX is really old, version 2, L^AT_EX can detect this, and sets `\@TeXversion` to 2 if it is not set in the `cfg` file.

2.1 INITEX

```

4  <*dircheck>
5  <*initex>
6  <initex> \ifnum\catcode`\'f=1
7  <initex>   \errmessage
8  <initex>   {LaTeX must be made using an initex with no format preloaded}
9  <initex> \fi
10 \catcode`\'f=1
11 \catcode`\'j=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!

```

12 \ifx\directlua\undefined
13 \else
14   \ifx\luatexversion\undefined

```

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

```

15   \directlua{tex.enableprimitives("",%
16             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.

The unprefixed forms are *not* undefined for improved compatibility with external packages when rolling back the format.

```

17 </initex>
18 </dircheck>
19 <*initex, latexrelease>
20 <latexrelease> \ifx\directlua\undefined\else
21 <latexrelease> \IncludeInRelease{2015/10/01}{\luatexluafunction}
22 <latexrelease>                               {LuaTeX (prefixed names)}%
23   \directlua{tex.enableprimitives("",%
24             tex.extraprimitives("omega", "aleph", "luatex"))}
25 <latexrelease> \EndIncludeInRelease
26 <latexrelease> \IncludeInRelease{0000/00/00}{\luatexluafunction}
27 <latexrelease>                               {LuaTeX (prefixed names)}%
28 <latexrelease> \directlua{
29 <latexrelease>   tex.enableprimitives(
30 <latexrelease>     "luatex",
31 <latexrelease>     tex.extraprimitives("core", "omega", "aleph", "luatex")
32 <latexrelease>   )
33 <latexrelease> }
34 <latexrelease> \EndIncludeInRelease
35 <latexrelease> \fi
36 </initex, latexrelease>
37 <*dircheck>
38 <*initex>
39   \fi
40 \fi

```

A test can now be made for eTeX.

```

41 <initex> \ifx\etexversion\undefined
42 <initex>   \errmessage
43 <initex>   {LaTeX requires e-TeX}
44 <initex>   \expandafter\endinput

```

```

45  <initex>\fi
    That distraction over, back to the basics of a format.
46  \catcode`#=6
47  \catcode`^=7
48  \chardef\active=13
49  \catcode`\@=11
50  \countdef\count@=255
51  \let\bgroup={ \let\egroup=}
52  \ifx\@input\undefined\let\@input\input\fi
53  \ifx\@end\undefined\let\@end\end\fi
54  \chardef\@inputcheck0
55  \chardef\sixt@n=16
56  \newlinechar`^^J
57  \def\typeout{\immediate\write17}
58  \def\dospecials{\do\ \do\\\do{\do}\}\do$\do\&%
59  \do#\do`^{\do_}\do%\do`~}
60  \def\@makeother#1{\catcode`#1=12\relax}
61  \def\space{ }
62  \def\@tempswafalse{\let\if@tempswa\iffalse}
63  \def\@tempswatrue{\let\if@tempswa\iftrue}
64  \let\if@tempswa\iffalse
65  \def\loop#1\repeat{\def\iterate{\#1\relax\expandafter\iterate\fi}%
66  \iterate\let\iterate\relax}
67  \let\repeat\fi
68  
```

2.2 Some bits of 2e

```

69  <*2ekernel>
70  \def\two@digits#1{\ifnum#1<10 0\fi\number#1}
71  \long\def\@firstoftwo#1#2{#1}
72  \long\def\@secondoftwo#1#2{#2}

```

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

```

73  \def\ProvidesFile#1{%
74    \begingroup
75      \catcode` 10 %
76      \ifnum \endlinechar<256 %
77        \ifnum \endlinechar>\m@ne
78          \catcode\endlinechar 10 %
79        \fi
80      \fi
81      \@makeother`%
82      \@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}}
83      \def\@providesfile#1[#2]{%
84        \wlog{File: #1 #2}%
85        \@addtofilelist{ #2}%
86        \endgroup
87      \long\def\@addtofilelist#1{%
88        \def\@empty{}%
89        \catcode`\%=12
90        \def\@percentchar{%
91          \catcode`\%=14
92          \let\@currdir\@undefined
93          \let\input@path\@undefined

```

```

94 \let\filename@parse\@undefined
\strip@prefix
95 \def\strip@prefix#1>{ }
96 </2ekernel>
(End of definition for \strip@prefix.)

```

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.

```

97 {*docstrip}
98 \openin15=texsys.cfg
99 \ifeof15
100 \typeout{** Writing a default texsys.cfg}
101 \immediate\openout15=texsys.cfg
102 \begingroup
103 \catcode`^=M\active%
104 \let^=M\par%
105 \def\reserved@a#1^=M{%
106 \def\reserved@b{#1}%
107 \ifx\reserved@b\reserved@c\endgroup\else%
108 \immediate\write15{#1}%
109 \expandafter\reserved@a\fi}%
110 \def\reserved@d#1START^=M{\let\do\@makeother\dospecials\reserved@a}%
111 \catcode`%12
112 \def\reserved@c{END}%
113 \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 `\langle dir \rangle` is an entry in the input path, TeX will try to load the expansion of `\langle dir \rangle\langle filename \rangle\langle space \rangle`

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

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

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

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

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

`\@TeXversion` You should not need to set this macro in `texsys.cfg`. L^AT_EX 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.

114 `%\def\@currdir{./}`
115 `%\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 `/`.

```
116 % \def\@currdir{./}
117 % \def\input@path{%
118 %   {/usr/local/lib/tex/inputs/distrib/}%
119 %   {/usr/local/lib/tex/inputs/contrib/}%
120 %   {/usr/local/lib/tex/inputs/local/}%
121 % }
```

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.

```
122 % \def\@currdir{./}
123 % \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.

```
124 % \def\@currdir{./}
125 % \def\input@path{%
126 %   {c:/tex/inputs/distrib/}%
127 %   {c:/tex/inputs/contrib/}%
128 %   {c:/tex/inputs/local/}%
129 % }
```

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.

```
130 % \def\@currdir{[]}
131 % \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:

```
132 % \def\@currdir{[]}
133 % \def\input@path{%
134 %   {tex_inputs:}%
135 %   {SOMEDISK:[SOME.TEX.DIRECTORY]}%
136 % }
```

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.

```
137 % \def\@currdir{:  
138 % \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.

```
139 % \def\@currdir{:  
140 % \def\input@path{  
141 %   {Hard-Disk:Applications:TeX:TeX-inputs:  
142 %   {Hard-Disk:Applications:TeX:My-inputs:  
143 % }
```

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:

```
144 % \def\@dir#1#2 {  
145 %   \@d@r{#1}#2..\@nil}  
146 % \def\@d@r#1#2.#3.#4\@nil{  
147 %   <\ifx\@dir#1\@dir\else#1\ifx\@dir#3\@dir\else.\fi\fi#3>#2 }  
148 %  
149 % \def\@currdir{\@dir{}}  
150 % \def\input@path{  
151 %   {\@dir{area.one}}%  
152 %   {\@dir{area.two}}%  
153 % }
```

END

```
154 \immediate\closeout15
```

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

```
155 \else  
156 \typeout{** Using the existing texsys.cfg}  
157 \closein15  
158 \input texsys.cfg  
159 \fi  
160 </docstrip>
```

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

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

4 Setting \@currdir

\@currdir
\IfFileExists This is a local definition of \IfFileExists. It tries to relocate `texssys.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.

```
162 \begingroup
163 \count@\time
164 \divide\count@ 60
165 \count2=-\count@
166 \multiply\count2 60
167 \advance\count2 \time
```

The current date and time stamp.

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

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

```
171 \immediate\openout15=texsys.aux
172 \immediate\write15{\today^J}
173 \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!

```
174 \def\IfFileExists#1#2#3{%
175   \openin\@inputcheck#1 %
176   \ifeof\@inputcheck
177     #3\relax
178   \else
179     \read\@inputcheck to \reserved@a
180     \ifx\reserved@a\today
181       \typeout{#1 found}#2\relax
182     \else
183       \typeout{BAD: old file \reserved@a (should be \today)}%
184       #3\relax
185     \fi
186   \fi
187   \closein\@inputcheck}
188 \endlinechar=-1
```

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

```
189 \ifx\@currdir\@undefined
190   \IfFileExists{./texsys.aux}{\gdef\@currdir{./}}%
191   {\IfFileExists{}{texsys.aux}{\gdef\@currdir{[]}}%
192   {\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.

```

193  \ifx\@currdir\@undefined
194    \global\let\@currdir\@empty
195    \typeout{^^J^^J%
196      !! No syntax for the current directory could be found^^J%
197    }%
198  \fi

```

Otherwise `\@currdir` was defined in `texsys.cfg`. In this case check that the syntax specified works on this system. (In case a complete L^AT_EX system has been copied from one system to another.) If the test fails, give up. The installer should remove or correct the offending `texsys.cfg` and try again.

```

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

```

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

```

207 \fi
208 \immediate\closeout15 %
209 \endgroup
210 \typeout{^^J^^J%
211   \noexpand\@currdir set to:
212   \expandafter\strip@prefix\meaning\@currdir.^^J%
213 }

```

(End of definition for `\@currdir`, `\IfFileExists`, and `\today`.)

Stop here if the file is being used unstripped.

```

214 <*docstrip>
215 \relax\endinput
216 </docstrip>

```

5 Setting `\input@path`

Earlier versions of this file attempted to automatically test whether `\input@path` was required, and interactively prompt for a path if necessary. This was not found to be very reliable. The first-time installer of L^AT_EX 2_E can not be expected to have enough information to supply the correct information to the prompts. Now the interaction is omitted. After the format is made the installer can attempt to run the test document `ltxcheck.tex` through L^AT_EX 2_E. This will check, among 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.

```

217 \typeout{^^J%

```

```

218     Assuming \noexpand\openin and \noexpand\input^^J%
219     \ifx\input@path\@undefined
220         have the same search path.^^J%
221     \else
222         \input@path has been defined in texsys.cfg.
223             have different search paths.^^J%
224             LaTeX will use the path specified by \noexpand\input@path:^^J%
225         \fi
226     \}

```

(End of definition for \input@path.)

6 Filename Parsing

\filename@parse Split a filename into its components.

```

226 \ifx\filename@parse\@undefined
227   \def\reserved@a{./}\ifx\@currdir\reserved@a
228 \filename@parse was not specified in texsys.cfg, but \@currdir looks like UNIX...
229   \typeout{^^JDefining UNIX/DOS style filename parser.^^J}
230   \def\filename@parse#1{%
231     \let\filename@area\empty
232     \expandafter\filename@path#1/\\"}

```

Search for the last /.

```

233   \def\filename@path#1/#2\\{%
234     \def\reserved@a{\filename@simple#1.\\"}%
235   \else
236     \edef\filename@area{\filename@area#1}%
237     \def\reserved@a{\filename@path#2\\"}%
238   \fi
239 \reserved@a}

```

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

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

```

241   \typeout{^^JDefining VMS style filename parser.^^J}
242   \def\filename@parse#1{%
243     \let\filename@area\empty
244     \expandafter\filename@path#1\\"}

```

Search for the last].

```

245   \def\filename@path#1]#2\\{%
246     \def\reserved@a{\filename@simple#1.\\"}%
247   \else
248     \edef\filename@area{\filename@area#1}%
249     \def\reserved@a{\filename@path#2\\"}%
250   \fi
251 \reserved@a}

```

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

```

\filename@parse was not specified in texsys.cfg, but \@currdir looks like Macintosh...
254   \typeout{^^JDefining Mac style filename parser.^^J}
255   \def\filename@parse#1{%
256     \let\filename@area@\empty
257     \expandafter\filename@path#1:\\}
258 
259 Search for the last :.
260 
261   \def\filename@path#1:#2\\{%
262     \ifx\#2\\%
263       \def\reserved@a{\filename@simple#1.\\}%
264     \else
265       \edef\filename@area{\filename@area#1:}%
266       \def\reserved@a{\filename@path#2\\}%
267     \fi
268     \reserved@a
269   }
270 
271 \else
272 
273 \filename@parse was not specified in texsys.cfg. So just make a simple parser that
274 always sets \filename@area to empty.
275 
276   \typeout{^^JDefining generic filename parser.^^J}
277   \def\filename@parse#1{%
278     \let\filename@area@\empty
279     \expandafter\filename@simple#1.\\}
280 
281 \fi\fi\fi
282 
283 \filename@simple is used by all three versions. Finally we can split off the extension.
284 
285   </dircheck>
286   <*dircheck, latexrelease>
287   <texreleas>\IncludeInRelease{2019/10/01}{\filename@simple}
288   <texreleas>                                {Final dot for extension}%
289   \def\filename@simple#1.#2\\{%
290     \ifx\#2\\%
291       \let\filename@ext\relax
292       \edef\filename@base{#1}%
293     \else
294       \filename@dots{#1}#2\\%
295     \fi}
296 
297   \def\filename@dots#1#2.#3\\{%
298     \ifx\#3\\%
299       \def\filename@ext{#2}%
300       \edef\filename@base{#1}%
301     \else
302       \filename@dots{#1.#2}#3\\%
303     \fi}
304 
305   <texreleas>\EndIncludeInRelease
306   <texreleas>\IncludeInRelease{0000/00/00}{\filename@simple}
307   <texreleas>                                {Final dot for extension}%
308   <texreleas> \def\filename@simple#1.#2\\{%
309     \ifx\#2\\%
310       \let\filename@ext\relax
311       \else
312         \edef\filename@ext{\filename@dots#2\\}%
313     \fi}

```

```

298 <{latexrelease}>      \fi
299 <{latexrelease}>      \edef\filename@base{\#1}%
300 <{latexrelease}> \EndIncludeInRelease
301 </dircheck, latexrelease>
302 <{*dircheck}>

        Remove a final dot, added earlier.

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

304 \else

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

305 \typeout{^^J^^J%
306   \noexpand\filename@parse was defined in texsys.cfg:^^J%
307   \expandafter\strip@prefix\meaning\filename@parse.^^J%
308 }
309 \fi

(End of definition for \filename@parse.)

```

7 T_EX Versions

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

```

310 \ifx\@TeXversion\undefined
311   \ifx\@undefined\inputlineno
312     \def\@TeXversion{2}
313   \else
314     \catcode`\\=active
315     \def\reserved@a{\if#1\string^3\fi}
316     \edef\reserved@a{\expandafter\reserved@a\string^J\@C}
317     \ifx\reserved@a\empty\else\gdef\@TeXversion{3}\fi
318   \fi
319 \fi

(End of definition for \@TeXversion.)

320 %% ---- END temporary definitions for bootstrapping ----
321 </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 02

ltplain.dtx

1 Plain T_EX

L^AT_EX includes almost all of the functionality of Knuth's original 'Basic Macros'. That is, the plain T_EX format described in Appendix B of the T_EXBook. However, some of the user commands are not much use so, in order to save memory, we may remove them from the kernel into a package. Here is a list of the commands that may be removed (PROBABLY NOT COMPLETE).

```
\magstep     \magstephalf  
\mathhexbox  
\vglue      \vgl@  
\hglue      \hgl@
```

This file is by now very small as most of it has been moved to more appropriate kernel files: it may disappear completely one day.

L^AT_EX font definitions are done using NFSS2 so none of PLAIN's font definitions are in L^AT_EX.

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

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

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

We had to define the `\catcodes` right away, before the message line, since `\message` uses the { and } characters. When INITEX (the T_EX initializer) starts up, it has defined the following `\catcode` values:

```
\catcode`\^\^@=\=9 % ascii null is ignored  
\catcode`\^\^M=\=5 % ascii return is end-line  
\catcode`\\=\=0 % backslash is TeX escape character  
\catcode`%=\=14 % percent sign is comment character  
\catcode` \ =\=10 % ascii space is blank space  
\catcode`\^\^?=15 % ascii delete is invalid  
\catcode`A=\=11 ... \catcode`Z=\=11 % uppercase letters  
\catcode`a=\=11 ... \catcode`z=\=11 % lowercase letters  
all others are type 12 (other)
```

Here is a list of the characters that have been specially catcoded:

```
13 \def\dospecials{\do\ \do\\\do\{\do\}\do\$\\do\&%  
14 \do\#\do\^\do\_\\do\%\do\~}
```

(not counting ascii null, tab, linefeed, formfeed, return, delete) Each symbol in the list is preceded by , which can be defined if you want to do something to every item in the list.

We make @ signs act like letters, temporarily, to avoid conflict between user names and internal control sequences of plain format.

15 \catcode`@=11

To make the plain macros more efficient in time and space, several constant values are declared here as control sequences. If they were changed, anything could happen; so they are private symbols.

\@ne Small constants are defined using \chardef.

\tw@ 16 \chardef\@ne=1

\thr@@ 17 \chardef\tw@=2

\sixt@@n 18 \chardef\thr@@=3

\@cclv 19 \chardef\sixt@@n=16

20 \chardef\@cclv=255

(End of definition for \@ne and others.)

\@cclvi Constants above 255 defined using \mathchardef.

\@m 21 \mathchardef\@cclvi=256

\@M 22 \mathchardef\@m=1000

\@MM 23 \mathchardef\@M=10000

24 \mathchardef\@MM=20000

(End of definition for \@cclvi and others.)

Allocation of registers

Here are macros for the automatic allocation of \count, \box, \dimen, \skip, \muskip, and \toks registers, as well as \read and \write stream numbers, \fam codes, \language codes, and \insert numbers.

25 \message{registers,}

When a register is used only temporarily, it need not be allocated; grouping can be used, making the value previously in the register return after the close of the group. The main use of these macros is for registers that are defined by one macro and used by others, possibly at different nesting levels. All such registers should be defined through these macros; otherwise conflicts may occur, especially when two or more macro packages are being used at the same time.

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

The following counters are reserved:

0 to 9 page numbering

10 count allocation

11 dimen allocation

12 skip allocation

13 muskip allocation

14 box allocation

15 toks allocation

16 read file allocation

17 write file allocation

18 math family allocation

19 language allocation

20 insert allocation

21 the most recently allocated number

22 constant -1
End of historical L^AT_EX 2.09 comments.

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

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

1, 3, 5, 7, 9,

and always non-\global assignments with respect to registers

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

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

```

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

```

\insc@unt The insertion counter and most recent allocation.
\allocationnumber

```

37 \countdef\insc@unt=20
38 \countdef\allocationnumber=21

```

(*End of definition for \insc@unt and \allocationnumber.*)

\m@ne The constant -1.

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

(*End of definition for \m@ne.*)

\wlog Write on log file (only)

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

(*End of definition for \wlog.*)

\count@ Here are abbreviations for the names of scratch registers that don't need to be allocated.
\dimen@
\dimen@i
\dimen@ii
\skip@
\toks@

```
41 \countdef\count@=255
```

```
42 \dimendef\dimen@=0
```

```
43 \dimendef\dimen@i=1 % global only
```

```
44 \dimendef\dimen@ii=2
```

```
45 \skipdef\skip@=0
```

```
46 \toksdef\toks@=0
```

(End of definition for \count@ and others.)

```
\newcount Now, we define \newcount, \newbox, etc. so that you can say \newcount\foo and \foo  
\newdimen will be defined (with \countdef) to be the next counter.  
\newskip To find out which counter \foo is, you can look at \allocationnumber.  
\newmuskip Since there's no \boxdef command, \chardef is used to define a \newbox,  
\newbox, \newinsert, \newfam, and so on.  
\newtoks LATEX change: remove \outer from \newcount and \newdimen (FMi) This is nec-  
\newread essary to use \newcount inside \if... later on. Also remove from \newskip, \newbox  
\newwrite and \newfam (DPC) to save later redefinition.  
\newfam  
\newlanguage 47 </2ekernel>  
48 {*2ekernel | latexrelease}  
49 <| latexrelease> \IncludeInRelease{2015/01/01}%  
50 <| latexrelease> {\newcount}{Extended Allocation}%  
51 \def\newcount {\e@alloc\count \countdef {\count10}\insc@unt\flo@at@count}  
52 \def\newdimen {\e@alloc\dimen \dimendef {\count11}\insc@unt\flo@at@count}  
53 \def\newskip {\e@alloc\skip \skipdef {\count12}\insc@unt\flo@at@count}  
54 \def\newmuskip  
55 {\e@alloc\muskip\muskipdef{\count13}\m@ne\e@alloc@top}
```

For compatibility use \chardef in the classical range.

```
56 \def\newbox {\e@alloc\box  
57 {\ifnum\allocationnumber<\@ccclvi  
58 \expandafter\chardef  
59 \else  
60 \expandafter\@alloc@chardef  
61 \fi}  
62 {\count14}\insc@unt\flo@at@count}  
63 \def\newtoks {\e@alloc\toks \toksdef{\count15}\m@ne\@alloc@top}  
64 \def\newread {\e@alloc\read \chardef{\count16}\m@ne\sixt@@n}  
Skip \write18 due to its traditional use as a shell-escape.  
65 \ifx\directlua\@undefined  
66 \def\newwrite {\e@alloc\write \chardef{\count17}\m@ne\sixt@@n}  
67 \else  
68 \def\newwrite {\e@alloc\write  
69 {\ifnum\allocationnumber=18  
70 \advance\count17\@ne  
71 \allocationnumber\count17 %  
72 \fi  
73 \global\chardef} %  
74 {\count17} %  
75 \m@ne  
76 {128}}  
77 \fi  
78 \def\new@mathgroup  
79 {\e@alloc\mathgroup\chardef{\count18}\m@ne\@mathgroup@top}  
80 \let\newfam\new@mathgroup  
81 \ifx\directlua\@undefined  
82 \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne\@ccclvi}  
83 \else  
84 \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne{16384}}
```

```

85 \fi
86 </2ekernel | latexrelease>
87 <latexrelease>\EndIncludeInRelease
88 <latexrelease>\IncludeInRelease{0000/00/00}%
89 <latexrelease>{\newcount}{Extended Allocation}%
90 <latexrelease>\def\newcount{\alloc@0\count\countdef\insc@unt}
91 <latexrelease>\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}
92 <latexrelease>\def\newskip{\alloc@2\skip\skipdef\insc@unt}
93 <latexrelease>\def\newmuskip{\alloc@3\muskip\muskipdef\ccclvi}
94 <latexrelease>\def\newbox{\alloc@4\box\chardef\insc@unt}
95 <latexrelease>\def\newtoks{\alloc@5\toks\toksdef\ccclvi}
96 <latexrelease>\def\newread{\alloc@6\read\chardef\sixt@n}
97 <latexrelease>\def\newwrite{\alloc@7\write\chardef\sixt@n}
98 <latexrelease>\def\new@mathgroup{\alloc@8\fam\chardef\sixt@n}
99 <latexrelease>\def\newlanguage{\alloc@9\language\chardef\ccclvi}
100 <latexrelease>\let\newfam\new@mathgroup
101 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\newcount` and others.)

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

```

102 <*2ekernel | latexrelease>
103 <latexrelease>\IncludeInRelease{2015/01/01}%
104 <latexrelease>{\e@alloc@chardef}{Extended Allocation}%
105 \ifx\directlua\undefined
106   \ifx\widowpenalties\undefined

```

classic tex has 2^8 registers.

```

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

```

etex and xetex have 2^{15} registers.

```

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

```

luatex has 2^{16} registers.

```

114   \chardef\e@alloc@top=65535
115   \let\e@alloc@chardef\chardef
116 \fi
117 </2ekernel | latexrelease>
118 <latexrelease>\EndIncludeInRelease
119 <latexrelease>\IncludeInRelease{0000/00/00}%
120 <latexrelease>{\e@alloc@chardef}{Extended Allocation}%
121 <latexrelease>\let\e@alloc@top\undefined
122 <latexrelease>\let\e@alloc@chardef\undefined
123 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\e@alloc@chardef` and `\e@alloc@top`.)

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

124 {*2ekernel | latexrelease}
125 <tex>\IncludeInRelease{2015/01/01}%
126 <tex>\fam{\e@mathgroup@top}{Extended Allocation}%
127 \ifx\Umathcode\undefined

```
- classic and e tex have 16 fam (0–15).
- ```

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

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

130 \chardef\mathgroup@top=256
131 \fi
132 <tex>/2ekernel | latexrelease
133 <tex>\EndIncludeInRelease
134 <tex>\IncludeInRelease{0000/00/00}%
135 <tex>\fam{\e@mathgroup@top}{Extended Allocation}%
136 <tex>\let\mathgroup@top\undefined
137 <tex>\EndIncludeInRelease

```
- (End of definition for \e@mathgroup@top.)
- \e@alloc A modified version of \alloc@ that takes the count register rather than just the final digit of its number (assuming \count1x). It also has an extra argument to give the top of the extended range.
- ```

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

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

138 {*2ekernel | latexrelease}
139 <tex>\IncludeInRelease{2015/01/01}{\e@alloc}{Extended Allocation}%
140 \def\@alloc#1#2#3#4#5#6{%
141 \global\advance#3\@ne
142 \e@ch@ck{#3}{#4}{#5}{#1}%
143 \allocationnumber#3\relax
144 \global#2#6\allocationnumber
145 \wlog{\string#6=\string#1\the\allocationnumber}}%
146 <tex>/2ekernel | latexrelease
147 <tex>\EndIncludeInRelease
148 <tex>\IncludeInRelease{0000/00/00}{\e@alloc}{Extended Allocation}%
149 <tex>\let\@alloc\undefined
150 <tex>\EndIncludeInRelease
151 {*2ekernel}

```
- (End of definition for \e@alloc.)
- \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 \c@freelist.

\extrafloats 152 {/2ekernel}
 {*2ekernel | latexrelease}
 {latexrelease} \IncludeInRelease{2015/10/01}
 {latexrelease} {\e@ch@ck}{Extended Allocation (checking)}%
156 \gdef\c@ch@ck#1#2#3#4{%
157 \ifnum#1<#2\else

```

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

```

158 \ifnum#1=#2\relax
159 \global\c@cclvi
160 \ifx\c@count\c@global\advance\c@count 10 \fi
161 \fi

```

Check we are below the extended limit.

```

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

```

```
\extrafloats 189 \ifx\c@numexpr\c@undefined
```

In classic TeX use \newinsert to allocate float boxes.

```

190 \def\extrafloats#1{%
191 \c@count@#1\relax
192 \ifnum\c@count@>\c@z@%
193 \newinsert\c@reserved@a
194 \c@global\expandafter\chardef

```

```

195 \csname bx@\the\allocationnumber\endcsname\allocationnumber
196 \@cons\@freelist{\csname bx@\the\allocationnumber\endcsname}%
197 \advance\count@\m@ne
198 \expandafter\extrafloats
199 \expandafter\count@
200 \fi
201 }%
202 \else

```

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

```

203 \def\extrafloats#1{%
204 \ifnum#1>\z@
205 \count@\numexpr\float@count-1\relax
206 \ifnum\count@<266 \ch@ck0\m@ne\insert\fi
207 \ch@ck0\count@\count
208 \ch@ck1\count@\dimen
209 \ch@ck2\count@\skip
210 \ch@ck4\count@\box
211 \global\edef\alloc@chardef\float@count\count@
212 \global\expandafter\edef\alloc@chardef
213 \csname bx@\the\float@count\endcsname\float@count
214 \@cons\@freelist{\csname bx@\the\float@count\endcsname}%
215 \expandafter\extrafloats\expandafter{\the\numexpr#1-1\expandafter}%
216 \fi}%
217 \fi
218 </2ekernel | latexrelease>
219 <latexrelease>\EndIncludeInRelease
220 <latexrelease>\IncludeInRelease{0000/00/00}%
221 <latexrelease> {\extrafloats}{Extra floats}%
222 <latexrelease>\let\float@count@undefined
223 <latexrelease>\let\extrafloats@undefined
224 <latexrelease>\EndIncludeInRelease
225 <*2ekernel>

```

(End of definition for `\e@ch@ck`, `\extrafloats`, and `\extrafloats`.)

`\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 definition gives spurious errors once the “classic” registers run out, so it is now defined to call `\e@alloc` internally.

```

226 </2ekernel>
227 <*2ekernel | latexrelease>
228 <latexrelease>\IncludeInRelease{2020/10/01}
229 <latexrelease> {\alloc@}{emulate alloc@}%
230 \def\alloc@#1#2#3#4{\e@alloc#2#3{\count1#1}#4\float@count}
231 </2ekernel | latexrelease>
232 <latexrelease>\EndIncludeInRelease
233 <latexrelease>\IncludeInRelease{0000/00/00}%
234 <latexrelease> {\alloc@}{emulate alloc@}%
235 <latexrelease>\def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
236 <latexrelease> \ch@ck#1#4#2%
237 <latexrelease> \allocationnumber\count1#1%

```

```

238 ⟨latexrelease⟩ \global#3#5\allocationnumber
239 ⟨latexrelease⟩ \wlog{\string#5=\string#2\the\allocationnumber}
240 ⟨latexrelease⟩\EndIncludeInRelease
241 ⟨*2ekernel⟩

```

(End of definition for `\alloc@`.)

`\newinsert`

```

242 ⟨/2ekernel⟩
243 ⟨*2ekernel | latexrelease⟩
244 ⟨latexrelease⟩\IncludeInRelease{2015/10/01}
245 ⟨latexrelease⟩ {\newinsert}{Extended \newinsert}%
246 \ifx\numexpr\undefined

```

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

```

247 \def\newinsert#1{\global\advance\insc@unt \m@ne
248 \ch@ck0\insc@unt\count
249 \ch@ck1\insc@unt\dimen
250 \ch@ck2\insc@unt\skip
251 \ch@ck4\insc@unt\box
252 \allocationnumber\insc@unt
253 \global\chardef#1\allocationnumber
254 \wlog{\string#1=\string\insert\the\allocationnumber}%
255 \else

```

The highest register allowed with `\insert`.

```

256 \ifx\directlua\undefined
257 \chardef\@insert@top255
258 \else
259 \chardef\@insert@top\@alloc@top
260 \fi

```

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

```

261 \def\newinsert#1{%
262 \tempswafalse
263 \global\advance\insc@unt\m@ne
264 \ifnum\count10<\insc@unt
265 \ifnum\count11<\insc@unt
266 \ifnum\count12<\insc@unt
267 \ifnum\count14<\insc@unt
268 \tempswatrue
269 \fi\fi\fi\fi
270 \if@tempswa
271 \allocationnumber\insc@unt
272 \else
273 \global\advance\insc@unt\@ne
274 \extrafloats\@ne
275 \next@currbox@\freelist
276 {\ifnum@\currbox<\@insert@top
277 \allocationnumber@\currbox
278 \else
279 \ch@ck0\m@ne\insert
280 \fi}%

```

```

281 {\ch@ck0\m@ne\insert}%
282 \fi
283 \global\chardef#1\allocationnumber
284 \wlog{\string#1=\string\insert\the\allocationnumber}%
285 }

286 \fi
287 {/2ekernel | latexrelease}

288 \EndIncludeInRelease
289 \IncludeInRelease{0000/00/00}%
290 \newinsert}{Extended \newinsert}%
291 \let\e@insert@top\@undefined
292 \def\newinsert#1{\global\advance\insc@unt \m@ne
293 \ch@ck0\insc@unt\count
294 \ch@ck1\insc@unt\dimen
295 \ch@ck2\insc@unt\skip
296 \ch@ck4\insc@unt\box
297 \allocationnumber\insc@unt
298 \global\chardef#1\allocationnumber
299 \wlog{\string#1=\string\insert\the\allocationnumber}}
300 \EndIncludeInRelease
301 {*2ekernel}

```

*(End of definition for \newinsert.)*

```

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

```

*(End of definition for \ch@ck.)*

```

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

```

*(End of definition for \newhelp.)*

**\@inputcheck** Allocate read stream for testing and output stream that is never open an thus writes to the terminal.

```

307 \newread\@inputcheck
308 \newwrite\@unused

```

*(End of definition for \@inputcheck and \@unused.)*

**\maxdimen** Here are some examples of allocation.

```

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

```

*(End of definition for \maxdimen and \hideskip.)*

```

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

```

(End of definition for \p@ and others.)

Assign initial values to TeX's parameters

315 \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.

```
316 \pretolerance=100
317 \tolerance=200 % INITEX sets this to 10000
318 \hbadness=1000
319 \vbadness=1000
320 \linepenalty=10
321 \hyphenpenalty=50
322 \exhyphenpenalty=50
323 \binoppenalty=700
324 \relpenalty=500
325 \clubpenalty=150
326 \widowpenalty=150
327 \displaywidowpenalty=50
328 \brokenpenalty=100
329 \predisplaypenalty=10000

330 % \postdisplaypenalty=0
331 % \interlinepenalty=0
332 % \floatingpenalty=0, set during \insert
333 % \outputpenalty=0, set before TeX enters \output
334 \doublehyphendemerits=10000
335 \finalhyphendemerits=5000
336 \adjdemerits=10000

337 % \looseness=0, cleared by TeX after each paragraph
338 % \pausing=0
339 % \holdinginserts=0
340 % \tracingonline=0
341 % \tracingmacros=0
342 % \tracingstats=0
343 % \tracingparagraphs=0
344 % \tracingpages=0
345 % \tracingoutput=0
```

In the past L<sup>A</sup>T<sub>E</sub>X used the default value of 1 for \tracinglostchars because this was the best it could do. This way one would at least get a warning in the .log file. e-T<sub>E</sub>X improved on that and supported a value of 2 to show the warning on the terminal, so we could have changed the default when we made the e-T<sub>E</sub>X extensions required—however, we overlooked that opportunity. In 2021 this parameter was improved on again and now also accepts the value 3 (error on the terminal). This made us realize that we should change the default. Using 3 would really be the best, but for compatibility reasons we only use 2.

```
346 \tracinglostchars=2
347 % \tracingcommands=0
348 % \tracingrestores=0
```

\tracingstacklevels For LuaT<sub>E</sub>X, the \tracingstacklevels functionality was implemented as a callback, so here we just define the count register to hold the value of the parameter.

```

349 </2ekernel>
350 <*2ekernel | latexrelease>
351 <latexrelease>\IncludeInRelease{2021/06/01}{\tracingstacklevels}%
352 <latexrelease> {\tracingstacklevels}%
353 \ifx\directlua\@undefined
354 % \tracingstacklevels=0 % added in 2021
355 \else
356 \newcount\tracingstacklevels
357 % Code for \tracingstacklevels defined in ltfinal.dtx
358 \fi
359 <latexrelease>\EndIncludeInRelease
360 <latexrelease>
361 <latexrelease>\IncludeInRelease{0000/00/00}{\tracingstacklevels}%
362 <latexrelease> {\tracingstacklevels}%
363 <latexrelease>\ifx\directlua\@undefined
364 <latexrelease>\else
365 <latexrelease> \let\tracingstacklevels\@undefined
366 <latexrelease>\fi
367 <latexrelease>\EndIncludeInRelease
368 </2ekernel | latexrelease>
369 <*2ekernel>

(End of definition for \tracingstacklevels.)

370 \uchyph=1
371 % \lefthyphenmin=2 \righthypenmin=3 set below
372 % \globaldefs=0
373 % \maxdeadcycles=25 % INITEX does this
374 % \hangafter=1 % INITEX does this, also TeX after each paragraph
375 % \fam=0
376 % \mag=1000 % INITEX does this
377 % \escapechar='\\ % INITEX does this
378 \defaulthyphenchar='-
379 \defaultskewchar=-1
380 % \endlinechar='\^M % INITEX does this
381 % \newlinechar=-1 \LaTeX\ sets this in ltdefns.dtx.
382 \delimiterfactor=901
383 % \time=now % TeX does this at beginning of job
384 % \day=now % TeX does this at beginning of job
385 % \month=now % TeX does this at beginning of job
386 % \year=now % TeX does this at beginning of job

```

In L<sup>A</sup>T<sub>E</sub>X we don't want box information in the transcript unless we do a full tracing.

```

387 \showboxbreadth=-1
388 \showboxdepth=-1
389 \errorcontextlines=-1
390 \hfuzz=0.1pt
391 \vfuzz=0.1pt
392 \overfullrule=5pt
393 \maxdepth=4pt
394 \splitmaxdepth=\maxdimen
395 \boxmaxdepth=\maxdimen

```

```

396 % \lineskiplimit=0pt, changed by \normalbaselines
397 \delimitershortfall=5pt
398 \nulldelimiterspace=1.2pt
399 \scriptspace=0.5pt
400 % \mathsurround=0pt
401 % \predisplaysize=0pt, set before TeX enters $$
402 % \displaywidth=0pt, set before TeX enters $$
403 % \displayindent=0pt, set before TeX enters $$
404 \parindent=20pt
405 % \hangindent=0pt, zeroed by TeX after each paragraph
406 % \hoffset=0pt
407 % \voffset=0pt
408 %
409 % \baselineskip=0pt, changed by \normalbaselines
410 % \lineskip=0pt, changed by \normalbaselines
411 \parskip=0pt plus 1pt
412 \abovedisplayskip=12pt plus 3pt minus 9pt
413 \abovedisplayshortskip=0pt plus 3pt
414 \belowdisplayskip=12pt plus 3pt minus 9pt
415 \belowdisplayshortskip=7pt plus 3pt minus 4pt
416 % \leftskip=0pt
417 % \rightskip=0pt
418 \topskip=10pt
419 \splittopskip=10pt
420 % \tabskip=0pt
421 % \spaceskip=0pt
422 % \xspaceskip=0pt
423 \parfillskip=0pt plus 1fil

```

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

```

424 \newskip\normalbaselineskip \normalbaselineskip=12pt
425 \newskip\normallineskip \normallineskip=1pt
426 \newdimen\normallineskiplimit \normallineskiplimit=0pt

```

(End of definition for \normalbaselineskip, \normallineskip, and \normallineskiplimit.)

\interfootlinepenalty

```

427 \newcount\interfootnotelinepenalty \interfootnotelinepenalty=100

```

(End of definition for \interfootlinepenalty.)

Definitions for preloaded fonts

\magstephalf

\magstep

```

428 \def\magstephalf{1095 }
429 \def\magstep#1{\ifcase#1 \or 1200\or 1440\or 1728\or
430 2074\or 2488\fi\relax}

```

(End of definition for \magstephalf and \magstep.)

Macros for setting ordinary text

```

\frenchspacing
\nonfrenchspacing 431 \def\frenchspacing{\sfcode`\. \sfcode`?\! \sfcode`\!\!`
432 \sfcode`\:\!` \sfcode`\;` \sfcode`\,,`
433 \def\nonfrenchspacing{\sfcode`\..3000\sfcode`?3000\sfcode`\!3000%`
434 \sfcode`\.:2000\sfcode`\;1500\sfcode`\,,1250 }

```

(End of definition for `\frenchspacing` and `\nonfrenchspacing`.)

`\normalbaselines`

```

435 \def\normalbaselines{\lineskip\normallineskip
436 \baselineskip\normalbaselineskip \lineskiplimit\normallineskiplimit}

```

(End of definition for `\normalbaselines`.)

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

`\I` 437 \def\^\M{\ } % control <return> = control <space>
438 \let\^\I\^\M % same for <tab>

(End of definition for `\M` and `\I`.)

```

\lq
\rq 439 \def\lq{`}
440 \def\rq{'}

```

(End of definition for `\lq` and `\rq`.)

`\lbrack`  
`\rbrack` 441 \def\lbrack{[]}
442 \def\rbrack{[]}

(End of definition for `\lbrack` and `\rbrack`.)

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

```

443 \def \aa {\r a}
444 \def \AA {\r A}

```

(End of definition for `\aa` and `\AA`.)

`\endgraf`

```

\endline 445 \let\endgraf=\par
446 \let\endline=\cr

```

(End of definition for `\endgraf` and `\endline`. These functions are documented on page 405.)

`\space`

```

447 \def\space{ }

```

(End of definition for `\space`.)

`\empty` This probably ought to go altogether, but let it to the L<sup>A</sup>T<sub>E</sub>X version to save space.

```

448 \let\empty\@empty

```

(End of definition for `\empty`.)

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

(End of definition for \null.)
```

```
\bgroup
\egroup
450 \let\bgroup=
451 \let\egroup=
```

(End of definition for \bgroup and \egroup.)

**\obeylines** In \obeylines, we say \let<sup>^^M</sup>=\obeyedline instead of \def<sup>^^M</sup>{\obeyedline} since this allows, for example, \let\obeyedline=\cr \obeylines \halign{....}

This is essentially a plain TeX trick and in its original version where you had to use to use \let\par=\cr not really a safe idea in L<sup>A</sup>T<sub>E</sub>X. If anybody used this trick this now breaks (and one needs to use \obeyedline instead).

```
452 </2ekernel>
453 <*2ekernel | latexrelease>
454 <latexrelease>\IncludeInRelease{2022/06/01}{\obeylines}%
455 <latexrelease> {Add a redirection to obeylines and obeyspaces}%
```

If the active <sup>^^M</sup> escapes, e.g., into a \write (which is effectively in a different context) then we don't want the definition from \obeylines but rather a simple \par (in fact even the primitive one, not the L<sup>A</sup>T<sub>E</sub>X version \para\_end: which is only defined later).

```
456 \begingroup
457 \catcode`^^M=\active % these lines must end with %
458 \gdef\obeylines{\catcode`^^M\active%
459 \let^^M=\obeyedline%
```

The next line ending the definition is rather curious and it took me awhile to understand why rollback fails. The problem is the following: if `latexrelease` is used, then blocks of `\IncludeInRelease ... \EndIncludeInRelease` are bypassed at high speed by grabbing each as a delimited argument. However, in that case <sup>^^M</sup> is seen not as code but as line ending characters and in that mode TeX discards everything from that point onwards to the real end of the line so it works like a comment — pretty strange really (and I think due to the fact that the original pascal compiler could have some garbage showing up after the normal line ending character). Thus we really have to make sure that any closing braces is not one the same line as an <sup>^^M</sup>, because otherwise it would get dropped and we end with unbalanced braces and never see the `\EndIncludeInRelease` — weird. In other places it doesn't matter because we aren't using the incomplete result.

```
460 }%
461 \global\let^^M\par % this is in case ^^M appears in a \write
462 \endgroup
```

**\obeyedline** The \obeyedline expands by default to \par with whatever definition \par has when it is executed. It can, however, be redefined (before calling \obeylines!) to achieve some special effects. If you want to alter this definition when already in the scope of \obeylines, it has no effect (because \let is used above). In that case simply make another call to \obeylines immediately. As you are in a restricted scope all that happens is that your redefinition is applied.

For the default definition we have to use \def not \let because the meaning of \par can change and we want to use the one that is current when \obeylines act.

There is a small subtlety here: in an `\edef` the active `^M` stayed put (because it was equal to the primitive `\par`), now `\obeyedline` expands and you get what it contains, i.e., in that case `\par`, into the `\edef` or `\mark` unless we use `\protected` on it.

```
463 \protected\gdef\obeyedline{\par}
```

The definition of `\obeyspaces` is changed in the same way and now executes `\obeyedspace` for each active space.

```
\obeyedspace
```

```
464 \global\let\obeyedspace\space
465 \begingroup
466 \catcode`\ =\active%
467 \gdef\obeyspaces{\catcode`\ \active\let =\obeyedspace}%
```

An active space elsewhere generates `\space` by default (for example in a `\write`).

```
468 \global\let =\space%
469 \endgroup
470 {/2ekernel | latexrelease}
471 <latexrelease>\EndIncludeInRelease
472 <latexrelease>\IncludeInRelease{0000/00/00}{\obeylines}%
473 <latexrelease> {Add a redirection to obeylines and obeyspaces}%
474 <latexrelease>
```

From 2019 onwards the commands are made robust (somewhat later in the kernel sources). So if we roll back they are robust, so when redefining them we have to get rid of the robust payload first. Otherwise that is seen by the later rollback below, which then installs a fragile version of the new definition on top of the one we roll back to here, sigh. `\kernel@make@fragile` also changes its definition (later own) so this is done directly.

```
475 <latexrelease>\expandafter\let\csname obeylines \endcsname@\undefined
476 <latexrelease>\expandafter\let\csname obeyspace \endcsname@\undefined
477 <latexrelease>
478 <latexrelease>\begingroup
479 <latexrelease>\catcode`^M=\active % these lines must end with %
480 <latexrelease> \gdef\obeyspaces{\catcode`^M\active \let^M\par %
```

Closing brace on a separate line (see comment above).

```
481 <latexrelease> }%
```

Another pitfall: if we do a rollback `\par` is no longer the primitive, so the roll back definition needs `\let` to what is now the primitive.

```
482 <latexrelease> \global\let^M\RawParEnd % this is in case ^M appears in a \write
483 <latexrelease>\endgroup
484 <latexrelease>\def\obeyspaces{\catcode`\ \active}
485 <latexrelease>
486 <latexrelease>\let\obeyedline@\undefined
487 <latexrelease>\let\obeyedspace@\undefined
488 <latexrelease>\EndIncludeInRelease
489 {*2ekernel}
```

*(End of definition for `\obeylines` and others.)*

`\loop` 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).

```
490 \long\def \loop #1\repeat{%
```

```

491 \def\iterate{\#1\relax % Extra \relax
492 \expandafter\iterate\fi
493 }
494 \iterate
495 \let\iterate\relax
496 }

```

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

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

*(End of definition for `\loop`, `\iterate`, and `\repeat`.)*

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

`\nointerlineskip`

`\offinterlineskip`

```

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

*(End of definition for `\nointerlineskip` and `\offinterlineskip`.)*

`\vglue`

`\hglue`

```

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

*(End of definition for `\vglue` and `\hglue`.)*

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).

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

*(End of definition for `\slash`.)*

`\break`

`\nobreak`

`\allowbreak`

```

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

*(End of definition for `\break`, `\nobreak`, and `\allowbreak`.)*

`\filbreak`

`\goodbreak`

```

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

*(End of definition for `\filbreak` and `\goodbreak`.)*

`\eject`

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

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

*(End of definition for `\eject`.)*

```

\removelastskip
514 \def\removelastskip{\ifdim\lastskip=\z@\else\vskip-\lastskip\fi}
(End of definition for \removelastskip.)
```

```

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

(End of definition for \smallbreak, \medbreak, and \bigbreak.)

```

\math
521 \def\math{\mathsurround\z@}
```

(End of definition for \math.)

**\underline** Due to L<sup>A</sup>T<sub>E</sub>X's redefinition of \underline plain T<sub>E</sub>X's \underline can be done in a simpler fashion (but do we need it at all?).

```

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

(End of definition for \underline.)

**\strutbox** L<sup>A</sup>T<sub>E</sub>X sets \strutbox in \set@fontsize.

```

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

(End of definition for \strutbox and \strut.)

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

```

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

(End of definition for \hidewidth.)

```

\narrower
526 \def\narrower{%
527 \advance\leftskip\parindent
528 \advance\rightskip\parindent}
```

(End of definition for \narrower.)

L<sup>A</sup>T<sub>E</sub>X defines \ae and similar commands elsewhere.

```

529 \chardef\%='\%
530 \chardef\&='\&
531 \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

```

532 \def\leavevmode{\unhbox\voidbox}
```

(End of definition for \leavevmode.)

```

\mathhexbox
533 \def\mathhexbox#1#2#3{\mbox{$\m@th \mathchar"#1#2#3$}}
(End of definition for \mathhexbox.)

\ialign
534 \def\ialign{\everycr{}\tabskip\z@skip\halign} % initialized \halign
(End of definition for \ialign.)

\oalign
\o@align
535 \def\oalign#1{\leavevmode\vtop{\baselineskip\z@skip \lineskip.25ex%
\o@align 536 \ialign{##\crcr#1\crcr}}}
537 \def\o@align{\lineskiplimit\z@ \oalign}
538 \def\ooalign{\lineskiplimit-\maxdimen \oalign}
(End of definition for \oalign, \o@align, and \ooalign.)

\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.
539 \def\sh@ft#1{\dimen0.00#1ex\multiply\dimen@{\fontdimen1\font
540 \kern-.0156\dimen0} % compensate for slant in lowered accents
(End of definition for \sh@ft.)

\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).
541 \def\ltx@sh@ft #1{%
542 \dimen@ #1%
543 \kern \strip@pt
544 \fontdimen1\font \dimen@
545 } % kern by #1 times the current slant
(End of definition for \ltx@sh@ft.)
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 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.
546 \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.
547 \def\dotfill{%
548 \leavevmode
549 \cleaders \hb@xt@ .44em{\hss.\hss}\hfill
550 \kern\z@}

```

(End of definition for \rulefill and \dotfill.)

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

551 `\sfcodes`)=0 \sfcodes`'=0 \sfcodes`\]=0`

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

Definitions related to output

`\magnification` doesn't work in LATEX.

```
def\magnification{\afterassignment\m@g\count@}
def\m@g{\mag\count@
\hsize6.5truein\vsiz8.9truein\dimen\footins8truein}
```

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

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

(End of definition for `\showoverfull`.)

```
\showoutput
\loggingoutput
553 \gdef\loggingoutput{\tracingoutput\@ne
554 \showboxbreadth\maxdimen\showboxdepth\maxdimen\errorstopmode}
555 \gdef\showoutput{\loggingoutput\showoverfull}
556 </2ekernel>
```

(End of definition for `\showoutput` and `\loggingoutput`.)

```
\tracingall
\loggingall
557 <|latexrelease|\IncludeInRelease{2021/06/01}{\loggingall}
558 <|latexrelease| {\tracingstacklevels and \tracinglostchars=3}%
559 <|2ekernel | latexrelease>
560 \edef\loggingall{%
561 \tracingstats\tw@
562 \tracingpages\@ne
563 \tracinglostchars\thr@@
564 \tracingparagraphs\@ne
565 \tracinggroups\@ne
566 \tracingifs\@ne
567 \tracingscantokens\@ne
568 \tracingnesting\@ne
569 \errorcontextlines\maxdimen
570 \ifdefined\tracingstacklevels \tracingstacklevels\maxdimen \fi
571 \noexpand \loggingoutput
572 \tracingmacros\tw@
573 \tracingcommands\thr@@
574 \tracingrestores\@ne
575 \tracingassigns\@ne
576 }%
577 \def\tracingall{\showoverfull\loggingall}
578 </2ekernel | latexrelease>
579 <|latexrelease|\EndIncludeInRelease
580 <|latexrelease|>
581 <|latexrelease|\IncludeInRelease{2015/01/01}{\loggingall}{etex tracing}%
582 <|latexrelease|\ifx\tracingscantokens\undefined
583 <|latexrelease|\gdef\loggingall{%
```

```

584 〈latexrelease〉 \tracingstats\tw@

585 〈latexrelease〉 \tracingpages\@ne

586 〈latexrelease〉 \tracinglostchars\@ne

587 〈latexrelease〉 \tracingparagraphs\@ne

588 〈latexrelease〉 \errorcontextlines\maxdimen

589 〈latexrelease〉 \loggingoutput

590 〈latexrelease〉 \tracingmacros\tw@

591 〈latexrelease〉 \tracingcommands\tw@

592 〈latexrelease〉 \tracingrestores\@ne

593 〈latexrelease〉 }%

594 〈latexrelease〉\else

595 〈latexrelease〉\gdef\loggingall{%
596 〈latexrelease〉 \tracingstats\tw@

597 〈latexrelease〉 \tracingpages\@ne

598 〈latexrelease〉 \tracinglostchars\tw@

599 〈latexrelease〉 \tracingparagraphs\@ne

600 〈latexrelease〉 \tracinggroups\@ne

601 〈latexrelease〉 \tracingifs\@ne

602 〈latexrelease〉 \tracingscantokens\@ne

603 〈latexrelease〉 \tracingnesting\@ne

604 〈latexrelease〉 \errorcontextlines\maxdimen

605 〈latexrelease〉 \loggingoutput

606 〈latexrelease〉 \tracingmacros\tw@

607 〈latexrelease〉 \tracingcommands\thr@@

608 〈latexrelease〉 \tracingrestores\@ne

609 〈latexrelease〉 \tracingassigns\@ne

610 〈latexrelease〉}%
611 〈latexrelease〉\fi

612 〈latexrelease〉\gdef\tracingall{\showoverfull\loggingall}

613 〈latexrelease〉\EndIncludeInRelease

614 〈latexrelease〉

615 〈latexrelease〉\IncludeInRelease{0000/00/00}{\loggingall}{etex tracing}%
616 〈latexrelease〉\gdef\loggingall{\tracingcommands\tw@\tracingstats\tw@

617 〈latexrelease〉 \tracingpages\@ne\tracinglostchars\@ne

618 〈latexrelease〉 \tracingmacros\tw@\tracingparagraphs\@ne\tracingrestores\@ne

619 〈latexrelease〉 \errorcontextlines\maxdimen\loggingoutput}

620 〈latexrelease〉 \gdef\tracingall{\loggingall\showoverfull}

621 〈latexrelease〉\EndIncludeInRelease

```

(End of definition for \tracingall and \loggingall.)

```
\tracingnone
622 〈latexrelease〉\IncludeInRelease{2015/01/01}{\tracingnone}%
623 〈latexrelease〉 {turn off etex tracing}%
624 〈*2ekernel | latexrelease〉
625 \edef\tracingnone{%
626 \tracingassigns\z@

627 \tracingrestores\z@

628 \tracingonline\z@

629 \tracingcommands\z@

630 \showboxdepth\m@ne

631 \showboxbreadth\m@ne

632 \tracingoutput\z@

633 \errorcontextlines\m@ne

```

```

634 \ifdefinable\tracingstacklevels {\tracingstacklevels\z@ \fi
635 \tracingnesting\z@
636 \tracingscantokens\z@
637 \tracingifs\z@
638 \tracinggroups\z@
639 \tracingparagraphs\z@
640 \tracingmacros\z@

```

None really means go back to the L<sup>A</sup>T<sub>E</sub>X “default” and for \tracinglostchars this should therefore be 2 these days.

```

641 \tracinglostchars\tw@
642 \tracingpages\z@
643 \tracingstats\z@
644 }%
645 </2ekernel | latexrelease>
646 <latexrelease>\EndIncludeInRelease
647 <latexrelease>
648 <latexrelease>\IncludeInRelease{2015/01/01}{\tracingnone}%
649 <latexrelease> {turn off etex tracing}%
650 <latexrelease>\ifx\tracingscantokens\undefined
651 <latexrelease>\def\tracingnone{%
652 <latexrelease> \tracingonline\z@
653 <latexrelease> \tracingcommands\z@
654 <latexrelease> \showboxdepth\m@ne
655 <latexrelease> \showboxbreadth\m@ne
656 <latexrelease> \tracingoutput\z@
657 <latexrelease> \errorcontextlines\m@ne
658 <latexrelease> \tracingrestores\z@
659 <latexrelease> \tracingparagraphs\z@
660 <latexrelease> \tracingmacros\z@
661 <latexrelease> \tracinglostchars\@ne
662 <latexrelease> \tracingpages\z@
663 <latexrelease> \tracingstats\z@
664 <latexrelease>}%
665 <latexrelease>\else
666 <latexrelease>\def\tracingnone{%
667 <latexrelease> \tracingassigns\z@
668 <latexrelease> \tracingrestores\z@
669 <latexrelease> \tracingonline\z@
670 <latexrelease> \tracingcommands\z@
671 <latexrelease> \showboxdepth\m@ne
672 <latexrelease> \showboxbreadth\m@ne
673 <latexrelease> \tracingoutput\z@
674 <latexrelease> \errorcontextlines\m@ne
675 <latexrelease> \tracingnesting\z@
676 <latexrelease> \tracingscantokens\z@
677 <latexrelease> \tracingifs\z@
678 <latexrelease> \tracinggroups\z@
679 <latexrelease> \tracingparagraphs\z@
680 <latexrelease> \tracingmacros\z@
681 <latexrelease> \tracinglostchars\@ne
682 <latexrelease> \tracingpages\z@
683 <latexrelease> \tracingstats\z@
684 <latexrelease>}%

```

```

685 〈\latexrelease〉\fi
686 〈\latexrelease〉\EndIncludeInRelease
687 〈\latexrelease〉
688 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\tracingnone}%
689 〈\latexrelease〉 {turn off etex tracing}%
690 〈\latexrelease〉\let\tracingnone\@undefined
691 〈\latexrelease〉\EndIncludeInRelease

```

(*End of definition for \tracingnone.*)

#### \hideoutput

```

692 〈*2ekernel | \latexrelease〉
693 〈\latexrelease〉\IncludeInRelease{2015/01/01}{\hideoutput}%
694 〈\latexrelease〉 {hide output from tracing}%
695 \def\hideoutput{%
696 \tracingoutput\z@%
697 \showboxbreadth\m@ne%
698 \showboxdepth\m@ne%
699 \tracingonline\m@ne%
700 }%
701 〈\latexrelease〉\EndIncludeInRelease
702 〈\latexrelease〉
703 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\hideoutput}%
704 〈\latexrelease〉 {hide output from tracing}%
705 〈\latexrelease〉\let\hideoutput\@undefined
706 〈\latexrelease〉\EndIncludeInRelease
707 〈/2ekernel | \latexrelease〉

```

(*End of definition for \hideoutput.*)

**L**A**T**E**X** change: `\showhyphens` Defined later.

Punctuation affects the spacing.

```

708 〈*2ekernel〉
709 \nonfrenchspacing
710 〈/2ekernel〉

```

# File 03

## ltvers.dtx

### 1 Version Identification

First we identify the date and version number of this release of L<sup>A</sup>T<sub>E</sub>X, and set \everyjob so that it is printed at the start of every L<sup>A</sup>T<sub>E</sub>X 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.

```
1 {*2ekernel}
2 \def\fmtname{LaTeX2e}
3 \edef\fmtversion
4 {/2ekernel}
5 <texrelease>\edef\latexreleaseversion
6 {*2ekernel | latexrelease}
7 {2024-11-01}
8 {/2ekernel | latexrelease}
9 {*2ekernel}
10 \def\patch@level{0}
```

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

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

(End of definition for \fmtname and others.)

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

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

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

\count0 is now the age of this file in months. Take a generous definition of ‘year’ so this message is not generated too often.

```

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

```

We store release info in the toks \LaTeXReleaseInfo to be used in \everyjob but also when \end{document} is executed. Instead of using \typeout we use \show@release@info so that we can write to the log only by changing that to \wlog.

```

36 \newtoks\LaTeXReleaseInfo
37 \everyjob\expandafter{\the\everyjob\the\LaTeXReleaseInfo}
38 \let\show@release@info\typeout
39 \ifnum0\ifnum\patch@level=0 \ifx\development@branch@name\@empty 1\fi\fi>0 %
40 \LaTeXReleaseInfo\expandafter{\the\LaTeXReleaseInfo
41 \show@release@info{\fmtname\space <\fmtversion>}}
42 \immediate
43 \write16{\fmtname\space<\fmtversion>}
44 \else\ifnum\patch@level>0
45 \LaTeXReleaseInfo\expandafter{\the\LaTeXReleaseInfo
46 \show@release@info{\fmtname\space <\fmtversion> patch level \patch@level}}
47 \immediate
48 \write16{\fmtname\space <\fmtversion> patch level \patch@level}
49 \else
50 \LaTeXReleaseInfo\expandafter{\the\LaTeXReleaseInfo
51 \show@release@info{\fmtname\space <\fmtversion>
52 pre-release-\number-\patch@level\space
53 \ifx\development@branch@name\@undefined \else
54 \ifx\development@branch@name\@empty \else
55 \space (\development@branch@name\space branch)%
56 \fi
57 \fi
58 }}
59 \immediate
60 \write16{\fmtname\space <\fmtversion>
61 pre-release-\number-\patch@level\space
62 \ifx\development@branch@name\@undefined \else
63 \ifx\development@branch@name\@empty \else
64 \space (\development@branch@name\space branch)%
65 \fi
66 \fi
67 }

```

```

68 \fi
69 \fi
70 </2ekernel>

\IncludeInRelease
\EndIncludeInRelease
 @IncludeInRelease
 @IncludeInRelease@se
@gobble@IncludeInRelease
@check@IncludeInRelease

 \def\IncludeInRelease#1{%
 \if@includeinrelease
 \PackageError{latexrelease}{mis-matched \IncludeInRelease}%
 {There is an \string\EndIncludeRelease\space missing}%
 \elsefalse
 \fi
 \ifnum0%
 \ifx\new@moduledate\empty\else 1\fi
 \ifnum \expandafter\@parse@version#1//00@nil=0 1\fi
 =11
 \expandafter\@firstoftwo
 \else
 \expandafter\@secondoftwo
 \fi
 {\@finish@module@release{#1}}%
 {\@kernel@ifnextchar[%
 {\@IncludeInRelease{#1}}
 {\@IncludeInRelease{#1}[#1]}}}
 \def\finish@module@release#1#2#3{%
 \toks@{[#1] #3}%
 \begingroup
 \edef\x{\detokenize\expandafter{\new@modulename}}%
 \edef\y{\detokenize{#2}}%
 \expandafter\endgroup
 \ifx\x\y \else
 \@latex@error{\noexpand\IncludeInRelease dated #1 in a module is not
 allowed.\MessageBreak Use a date at least equal to \new@moduledate
 \space for complete rollback}\@ehd
 \fi
 \ifnum\expandafter\@parse@version\new@moduledate//00@nil
 >\expandafter\@parse@version\fmtversion//00@nil
 \GenericInfo{}{Applying: \the\toks@}%
 \else
 \GenericInfo{}{Skipping: \the\toks@}%
 \expandafter\gobble@finish@module@release
 \fi}
 \long\def\gobble@finish@module@release#1\EndModuleRelease{%
 \EndModuleRelease}

 If a specific date has not been specified in \textrm{latexrelease} use '#1'.

113 \def\@IncludeInRelease#1[#2]{\@IncludeInRelease{#2}}
114 \def\@IncludeInRelease#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 over the first patch because that is only really needed for rolling forward. So maybe one day ...

```

117 \ifnum\expandafter\@parse@version#1//00@nil
118 >\expandafter\@parse@version\fmtversion//00@nil
119 \GenericInfo{}{Skipping: \the\toks@}%
120 \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
121 \else
122 \GenericInfo{}{Applying: \the\toks@}%
123 \@includeinreleasetrue
124 \expandafter\let\csname\string#2+\currname+IIR\endcsname\empty
125 \fi
126 \else
127 \GenericInfo{}{Already applied: \the\toks@}%
128 \expandafter\@gobble@IncludeInRelease
129 \fi
130 }

131 \def\EndIncludeInRelease{%
132 \if@includeinrelease
133 \@includeinreleasefalse
134 \else
135 \PackageError{latexrelease}{mis-matched EndIncludeInRelease}{}%
136 \fi
137 \if@skipping@module
138 \expandafter\new@module@skip
139 \fi}

140 \long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{%
141 \@includeinreleasefalse
142 \@check@IncludeInRelease#1\IncludeInRelease\@check@IncludeInRelease
143 \@end@check@IncludeInRelease}

144 \long\def\@check@IncludeInRelease#1\IncludeInRelease
145 #2#3\@end@check@IncludeInRelease{%
146 \ifx\@check@IncludeInRelease#2\else
147 \PackageError{latexrelease}{skipped IncludeInRelease for tag \string#2}{}%
148 \fi
149 \if@skipping@module
150 \expandafter\new@module@skip
151 \fi}

```

(End of definition for `\IncludeInRelease` and others.)

## 1.1 Declaring an all-new module

|                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\if@skipping@module</code><br><code>\NewModuleRelease</code><br><code>\EndModuleRelease</code><br><code>\new@module@skip</code><br><code>\new@modulename</code><br><code>\new@moduledate</code> | When we have a whole new module, we can't roll back to a date where such module exists, otherwise hundreds of "command already defined" errors will pop up. But we can't skip it altogether either, because the module might have changes we still want applied, so a more detailed cherry-picking of code chunks have to be done. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

152 \let\if@skipping@module\iffalse
153 \def\@skipping@moduletrue{\let\if@skipping@module\iftrue}
154 \def\@skipping@modulefalse{\let\if@skipping@module\iffalse}

```

```

155 \let\new@modulename\empty
156 \let\new@moduledate\empty
157 \def\NewModuleRelease#1#2#3{%
158 \ifx\new@modulename\empty \else
159 \@latex@error{Nested \noexpand\NewModuleRelease forbidden.}\@ehd \fi
160 \edef\new@moduledate{\#1}%
161 \edef\new@modulename{\#2}%
162 \GenericInfo{}{BEGIN module: \new@modulename\space (\new@moduledate)}%
163 \GenericInfo{}{ \@spaces\@spaces\@spaces\space#3\@gobble}%
164 \ifnum\sourceLaTeXdate<%
165 \expandafter\@parse@version\new@moduledate//00\@nil\relax
166 \ifnum\expandafter\@parse@version\fmtversion//00\@nil<%
167 \expandafter\@parse@version\new@moduledate//00\@nil\relax
168 \GenericInfo{}{Skipping module \new@modulename}%
169 \expandafter\expandafter
170 \expandafter\gobble@finish@module@release
171 \else
172 \GenericInfo{}{Applying module \new@modulename}
173 \@skipping@modulefalse
174 \fi
175 \else
176 \GenericInfo{}{Skipping module \new@modulename}
177 \@skipping@moduletrue
178 \expandafter\new@module@skip
179 \fi}
180 \long\def\new@module@skip#1\IncludeInRelease{%
181 \long\def\reserved@a##1\EndModuleRelease{}%
182 \if\relax\detokenize\expandafter{\reserved@a#1{}{}}\EndModuleRelease\relax
183 \else
184 \@latex@error{Missing mandatory \string\IncludeInRelease{0000/00/00}}\@ehc
185 \expandafter\@secondoftwo
186 \fi
187 \gobble
188 {\@expandtwoargs\IncludeInRelease
189 {0000/00/00}{\new@modulename}%
190 {ERROR! Emergency recovery}%
191 #1}%
192 \IncludeInRelease}
193 \def\EndModuleRelease{%
194 \ifx\new@modulename\empty
195 \@latex@error{Extra \string\EndModuleRelease.}\@eha
196 \else
197 \GenericInfo{}{END module: \new@modulename\space (\new@moduledate)}%
198 \let\new@modulename\empty
199 \let\new@moduledate\empty
200 \@skipping@modulefalse
201 \fi}

```

(End of definition for `\if@skipping@module` and others.)

202 `</2ekernel | latexrelease>`

# File 04

## 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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  kernel level plus as a loadable file which can be used with plain TeX and L<sup>A</sup>T<sub>E</sub>X.

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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  kernel from 2015 onward these registers are part of the block in the extended area reserved by the kernel (prior to 2015 the L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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 L<sup>A</sup>T<sub>E</sub>X format, however also extracted to the file `ltluatex.tex` which may be used with older L<sup>A</sup>T<sub>E</sub>X 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).

\newluacmd \newluacmd{<function>}
Like \newluafunction, but defines the command using \luadef instead of just assigning an integer.
```

```

\newprotectedluacmd \newluadef{\function}
 Like \newluacmd, but the defined command is not expandable.
\newwhatsit \newwhatsit{\whatsit}
 Defines a custom \whatsit, indexed from 1.
\newluabytocode \newluabytocode{\bytocode}
 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
 register (without backslash) into the lua.name table to be used in stack traces.
\catcodetable@initex Predefined category code tables with the obvious assignments. Note that the latex and
\catcodetable@string atletter tables set the full Unicode range to the codes predefined by the kernel.
\catcodetable@latex \setattribute{\attribute}{\value}
\catcodetable@atletter \unsetattribute{\attribute}
\setattribute Set and unset attributes in a manner analogous to \setlength. Note that attributes
\unsetattribute take a marker value when unset so this operation is distinct from setting the value to
zero.

```

### 3 Plain T<sub>E</sub>X interface

The `ltluatex` interface may be used with plain T<sub>E</sub>X using `\input{ltluatex}`. This inputs `ltluatex.tex` which inputs `etex.src` (or `etex.sty` if used with L<sup>A</sup>T<sub>E</sub>X) 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 T<sub>E</sub>X, 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 se-
 quence 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_bytocode luatexbase.new_bytocode(\bytocode)
 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.registernumer("#1") or "bad input")}%
}

\test{undefinedrubbish}

\test{space}

\test{hbox}

\test{@MM}

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

\test{strutbox}

\test{sixt@on}

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

If the demonstration code is processed with Lua<sup>L</sup>A<sub>T</sub>E<sub>X</sub> 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 L<sup>A</sup>T<sub>E</sub>X 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. If the `version` begins with a digit, a `v` will be added at the start in the log.

```
module_info luatexbase.module_info(<module>, <text>)
module_warning luatexbase.module_warning(<module>, <text>)
module_error luatexbase.module_error(<module>, <text>)
```

These functions are similar to L<sup>A</sup>T<sub>E</sub>X'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>, <type>, <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.

`declare_callback_rule luatexbase.declare_callback_rule(<name>, <first>, <relation>, <second>)` Adds an ordering constraint between two callback functions for callback `<name>`.  
The kind of constraint added depends on `<relation>`:

- before** The callback function with description `<first>` will be executed before the function with description `<second>`.
- after** The callback function with description `<first>` will be executed after the function with description `<second>`.
- incompatible-warning** When both a callback function with description `<first>` and with description `<second>` is registered, then a warning is printed when the callback is executed.
- incompatible-error** When both a callback function with description `<first>` and with description `<second>` is registered, then an error is printed when the callback is executed.
- unrelated** Any previously declared callback rule between `<first>` and `<second>` gets disabled.

Every call to `declare_callback_rule` with a specific callback `<name>` and descriptions `<first>` and `<second>` overwrites all previous calls with same callback and descriptions.

The callback functions do not have to be registered yet when the functions is called. Only the constraints for which both callback descriptions refer to callbacks registered at the time the callback is called will have an effect.

## 5 Implementation

<sup>1</sup> `(*2ekernel | tex | latexrelease)`  
<sup>2</sup> `{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’.

```

3 \langle latexrelease \rangle \IncludeInRelease{2015/10/01}
4 \langle latexrelease \rangle \newluafunction{LuaTeX}%
5 \ifnum\luatexversion<60 %
6 \wlog{*****}
7 \wlog{* LuaTeX version too old for ltluatex support *}
8 \wlog{*****}
9 \expandafter\endinput
10 \fi

```

Two simple L<sup>A</sup>T<sub>E</sub>X macros from `ltdefns.dtx` have to be defined here because `ltdefns.dtx` is not loaded yet when `ltluatex.dtx` is executed.

```

11 \long\def\@gobble#1{}
12 \long\def\@firstofone#1{#1}

```

## 5.2 Older L<sup>A</sup>T<sub>E</sub>X/Plain T<sub>E</sub>X setup

```
13 <*tex>
```

Older L<sup>A</sup>T<sub>E</sub>X formats don’t have the primitives with ‘native’ names: sort that out. If they already exist this will still be safe.

```

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

```

In pre-2014 L<sup>A</sup>T<sub>E</sub>X, or plain T<sub>E</sub>X, load `etex.{sty,src}`.

```

16 \ifx\documentclass\@undefined
17 \ifx\@alloc\@undefined
18 \input{etex.src}%
19 \fi
20 \catcode`\@=11 %
21 \outer\expandafter\def\csname newfam\endcsname
22 {\@alloc@8\fam\chardef\et@xmaxfam}%
23 \else
24 \RequirePackage{etex}
25 \expandafter\def\csname newfam\endcsname
26 {\@alloc@8\fam\chardef\et@xmaxfam}%
27 \expandafter\let\expandafter\new@mathgroup\csname newfam\endcsname
28 \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.

```

29 \edef\et@xmaxregs{\ifx\directlua\@undefined 32768\else 65536\fi}
luatex/xetex also allow more math fam.

```

```

30 \edef\et@xmaxfam{\ifx\Umathcode\@undefined\sixt@@n\else\@ccclvi\fi}
31 \count270=\et@xmaxregs % locally allocates \count registers
32 \count271=\et@xmaxregs % ditto for \dimen registers
33 \count272=\et@xmaxregs % ditto for \skip registers
34 \count273=\et@xmaxregs % ditto for \muskip registers
35 \count274=\et@xmaxregs % ditto for \box registers

```

```

36 \count 275=\et@xmaxregs % ditto for \toks registers
37 \count 276=\et@xmaxregs % ditto for \marks classes
 and 256 or 16 fam. (Done above due to plain/LATEX differences in ltluatex.)
38 % \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.

```

39 \expandafter\let\csname newcount\expandafter\expandafter\endcsname
40 \csname globcount\endcsname
41 \expandafter\let\csname newdimen\expandafter\expandafter\endcsname
42 \csname globdimen\endcsname
43 \expandafter\let\csname newskip\expandafter\expandafter\endcsname
44 \csname globskip\endcsname
45 \expandafter\let\csname newbox\expandafter\expandafter\endcsname
46 \csname globbox\endcsname

```

Define \e@alloc as in L<sup>A</sup>T<sub>E</sub>X (the existing macros in etex.src are hard to extend to further register types as they assume specific 26x and 27x count range). For compatibility the existing register allocation is not changed.

```

47 \chardef\e@alloc@top=65535
48 \let\@alloc@chardef\chardef
49 \def\@alloc#1#2#3#4#5#6{%
50 \global\advance#3\@ne
51 \e@ch@ck{#3}{#4}{#5}{#1%
52 \allocationnumber#3\relax
53 \global#2#6\allocationnumber
54 \wlog{\string#6=\string#1\the\allocationnumber}}%
55 \gdef\@ch@ck#1#2#3#4{%
56 \ifnum#1<#2\else
57 \ifnum#1=#2\relax
58 #1\@cclvi
59 \ifx\count#4\advance#1 10 \fi
60 \fi
61 \ifnum#1<#3\relax
62 \else
63 \errmessage{No room for a new \string#4}%
64 \fi
65 \fi}%

```

Fix up allocations not to clash with etex.src.

```

66 \expandafter\csname newcount\endcsname\@alloc@attribute@count
67 \expandafter\csname newcount\endcsname\@alloc@ccodetable@count
68 \expandafter\csname newcount\endcsname\@alloc@luafunction@count
69 \expandafter\csname newcount\endcsname\@alloc@whatsit@count
70 \expandafter\csname newcount\endcsname\@alloc@bytecode@count
71 \expandafter\csname newcount\endcsname\@alloc@luachunk@count

```

End of conditional setup for plain T<sub>E</sub>X / old L<sup>A</sup>T<sub>E</sub>X.

```

72 \fi
73 </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.

```
74 \ifx\@alloc@attribute@count\@undefined
75 \countdef\@alloc@attribute@count=258
76 \@alloc@attribute@count=\z@
77 \fi
78 \def\newattribute#1{%
79 \@alloc@attribute@attributedef
80 \@alloc@attribute@count\m@ne\@alloc@top#1%
81 }
```

(End of definition for \newattribute.)

- \setattribute Handy utilities.

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

(End of definition for \setattribute and \unsetattribute.)

### 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 initialization step. Table 0 is already taken (it's the global one for current use) so the allocation starts at 1.

```
84 \ifx\@alloc@ccodetable@count\@undefined
85 \countdef\@alloc@ccodetable@count=259
86 \@alloc@ccodetable@count=\z@
87 \fi
88 \def\newcatcodetable#1{%
89 \@alloc@catcodetable@chardef
90 \@alloc@ccodetable@count\m@ne{"8000}\#1%
91 \initcatcodetable\allocationnumber
92 }
```

(End of definition for \newcatcodetable.)

- \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.

```
93 \newcatcodetable\catcodetable@initex
94 \newcatcodetable\catcodetable@string
95 \begingroup
96 \def\setrangingcatcode#1#2#3{%
97 \ifnum#1>#2 %
98 \expandafter\@gobble
99 \else
100 \expandafter\@firstofone
101 \fi
102 {%
103 \catcode#1=#3 %
104 \expandafter\setrangingcatcode\expandafter
105 {\number\numexpr#1 + 1\relax}{#2}{#3}
106 }%
```

```

107 }
108 \catcodetable\catcodetable@initex
109 \catcode0=12 %
110 \catcode13=12 %
111 \catcode37=12 %
112 \setrangingcatcode{65}{90}{12}%
113 \setrangingcatcode{97}{122}{12}%
114 \catcode92=12 %
115 \catcode127=12 %
116 \savecatcodetable\catcodetable@string
117 \endgroup
118 }
119 }%
120 \newcatcodetable\catcodetable@latex
121 \newcatcodetable\catcodetable@atletter
122 \begingroup
123 \def\parseunicodedataI#1;#2;#3;#4\relax{%
124 \parseunicodedataII#1;#3;#2 First>\relax
125 }%
126 \def\parseunicodedataII#1;#2;#3 First>#4\relax{%
127 \ifx\relax#4\relax
128 \expandafter\parseunicodedataIII
129 \else
130 \expandafter\parseunicodedataIV
131 \fi
132 {#1}#2\relax%
133 }%
134 \def\parseunicodedataIII#1#2#3\relax{%
135 \ifnum 0%
136 \if L#21\fi
137 \if M#21\fi
138 >0 %
139 \catcode"#1=11 %
140 \fi
141 }%
142 \def\parseunicodedataIV#1#2#3\relax{%
143 \read\unicoderead to \unicodedataline
144 \if L#2%
145 \count0="#1 %
146 \expandafter\parseunicodedataV\unicodedataline\relax
147 \fi
148 }%
149 \def\parseunicodedataV#1;#2\relax{%
150 \loop
151 \unless\ifnum\count0>"#1 %
152 \catcode\count0=11 %
153 \advance\count0 by 1 %
154 \repeat
155 }%
156 \def\storedpar{\par}%
157 \chardef\unicoderead=\numexpr\count16 + 1\relax
158 \openin\unicoderead=UnicodeData.txt %
159 \loop\unless\ifeof\unicoderead %
160 \read\unicoderead to \unicodedataline

```

```

161 \unless\ifx\unicodedataline\storedpar
162 \expandafter\parseunicodedataI\unicodedataline\relax
163 \fi
164 \repeat
165 \closein\unicoderead
166 \@firstofone{%
167 \catcode64=12 %
168 \savecatcodetable\catcodetable@latex
169 \catcode64=11 %
170 \savecatcodetable\catcodetable@atletter
171 }
172 \endgroup

```

(End of definition for `\catcodetable@initex` and others.)

## 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.

```

173 \ifx\@alloc@luafunction@count\@undefined
174 \countdef\@alloc@luafunction@count=260
175 \@alloc@luafunction@count=\z@
176 \fi
177 \def\newluafunction{%
178 \@alloc@luafunction\@alloc@chardef
179 \@alloc@luafunction@count\m@ne\@alloc@top
180 }

```

(End of definition for `\newluafunction`.)

`\newluacmd` `\newprotectedluacmd` Additionally two variants are provided to make the passed control sequence call the function directly.

```

181 \def\newluacmd{%
182 \@alloc@luafunction\luadef
183 \@alloc@luafunction@count\m@ne\@alloc@top
184 }
185 \def\newprotectedluacmd{%
186 \@alloc@luafunction{\protected\luadef}
187 \@alloc@luafunction@count\m@ne\@alloc@top
188 }

```

(End of definition for `\newluacmd` and `\newprotectedluacmd`.)

## 5.6 Custom whatsits

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

```

189 \ifx\@alloc@whatst@count\@undefined
190 \countdef\@alloc@whatst@count=261
191 \@alloc@whatst@count=\z@
192 \fi
193 \def\newwhatst#1{%
194 \@alloc@whatst\@alloc@chardef
195 \@alloc@whatst@count\m@ne\@alloc@top#1%
196 }

```

(End of definition for \newwhatsit.)

## 5.7 Lua bytecode registers

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

```
197 \ifx\@alloc@bytecode@count\@undefined
198 \countdef\@alloc@bytecode@count=262
199 \@alloc@bytecode@count=\z@
200 \fi
201 \def\newluabytecode#1{%
202 \@alloc@luabytecode\@alloc@chardef
203 \@alloc@bytecode@count\m@ne\@alloc@top#1%
204 }
```

(End of definition for \newluabytecode.)

## 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.

```
205 \ifx\@alloc@luachunk@count\@undefined
206 \countdef\@alloc@luachunk@count=263
207 \@alloc@luachunk@count=\z@
208 \fi
209 \def\newluachunkname#1{%
210 \@alloc@luachunk\@alloc@chardef
211 \@alloc@luachunk@count\m@ne\@alloc@top#1%
212 {\escapechar\m@ne
213 \directlua{\lua.name[\the\allocationnumber]="\string#1"}}%
214 }
```

(End of definition for \newluachunkname.)

## 5.9 Lua loader

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

```
215 \def\now@and@everyjob#1{%
216 \everyjob\expandafter{\the\everyjob
217 #1%
218 }%
219 #1%
220 }
```

Load the Lua code at the start of every job. For the conversion of  $\text{\TeX}$  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.

```
221 <2ekernel> \now@and@everyjob{%
222 \begingroup
223 \attributedef\attributezero=0 %
224 \chardef\charzero=0 %
```

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

```

225 \countdef \CountZero =0 %
226 \dimendef \dimenzero =0 %
227 \mathchardef \mathcharzero =0 %
228 \muskipdef \muskipzero =0 %
229 \skipdef \skipzero =0 %
230 \toksdef \tokszero =0 %
231 \directlua{require("ltluatex")}
232 \endgroup
233 {2ekernel}
234 \textrun{\EndIncludeInRelease}

235 \textrun{\IncludeInRelease{0000/00/00}}
236 \textrun{\newluafunction{LuaTeX}%
237 \let\alloc@attribute@count\undefined
238 \let\newattribute\undefined
239 \let\setattribute\undefined
240 \let\unsetattribute\undefined
241 \let\alloc@ccodetable@count\undefined
242 \let\newcatcodetable\undefined
243 \let\catcodetable@initex\undefined
244 \let\catcodetable@string\undefined
245 \let\catcodetable@latex\undefined
246 \let\catcodetable@atletter\undefined
247 \let\alloc@luafunction@count\undefined
248 \let\newluafunction\undefined
249 \let\alloc@luafunction@count\undefined
250 \let\newwhatsit\undefined
251 \let\alloc@whatsit@count\undefined
252 \let\newluabytecode\undefined
253 \let\alloc@bytecode@count\undefined
254 \let\newluachunkname\undefined
255 \let\alloc@luachunk@count\undefined
256 \directlua{luatexbase.uninstall()}
257 \textrun{\EndIncludeInRelease}

```

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

```

258 \textrun{\IncludeInRelease{2017/01/01}%
259 \textrun{\fontencoding{TU} in everyjob}%
260 \textrun{\fontencoding{TU}\let\encodingdefault\f@encoding
261 \textrun{\ifx\directlua\undefined\else
262 {2ekernel}\everyjob\expandafter{%
263 {2ekernel} \the\everyjob
264 {*2ekernel,luatexrelease}
265 \directluat%
266 if xpcall(function ()%
267 require('luaotfloat-main')%
268 end,texio.write_nl) then %
269 local _void = luaotfloat.main ()%
270 else %
271 texio.write_nl('Error in luaotfloat: reverting to OT1')%
272 tex.print('\string\\def\string\\encodingdefault{OT1}')%
273 end %
274 }%

```

```

275 \let\f@encoding\encodingdefault
276 \expandafter\let\csname ver@luaotfloat.sty\endcsname\fmtversion
277 </2ekernel, latexrelease>
278 <latexrelease>\fi
279 <2ekernel> }
280 <latexrelease>\EndIncludeInRelease
281 <latexrelease>\IncludeInRelease{0000/00/00}%
282 <latexrelease> {\fontencoding}{TU in everyjob}%
283 <latexrelease>\fontencoding{OT1}\let\encodingdefault\f@encoding
284 <latexrelease>\EndIncludeInRelease
285 <2ekernel | latexrelease>\fi
286 </2ekernel | tex | latexrelease>

```

## 5.10 Lua module preliminaries

287 `(*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.

```

288 luatexbase = luatexbase or { }
289 local luatexbase = luatexbase

```

(*End of definition for luatexbase.*)

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

```

290 local string_gsub = string.gsub
291 local tex_count = tex.count
292 local tex_setcount = tex.setcount
293 local texio_write_nl = texio.write_nl
294 local flush_list = node.flush_list

295 local luatexbase_warning
296 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.

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

(*End of definition for modules.*)

`provides_module` Local function to write to the log.

```

298 local function luatexbase_log(text)
299 texio_write_nl("log", text)
300 end

```

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

```

301 local function provides_module(info)
302 if not (info and info.name) then
303 luatexbase_error("Missing module name for provides_module")
304 end
305 local function spaced(text)
306 return text and (" " .. text) or ""
307 end
308 luatexbase_log(
309 "Lua module: " .. info.name
310 .. spaced(info.date)
311 .. spaced(info.version and string.gsub(info.version or "", "%d", "v%1"))
312 .. spaced(info.description)
313)
314 modules[info.name] = info
315 end
316 luatexbase.provides_module = provides_module

```

*(End of definition for `provides_module`.)*

### 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.

```

317 local function msg_format(mod, msg_type, text)
318 local leader = ""
319 local cont
320 local first_head
321 if mod == "LaTeX" then
322 cont = string.gsub(leader, ".", " ")
323 first_head = leader .. "LaTeX: "
324 else
325 first_head = leader .. "Module " .. msg_type
326 cont = "(" .. mod .. ")"
327 .. string.gsub(first_head, ".", " ")
328 first_head = leader .. "Module " .. mod .. " " .. msg_type .. ":" ..
329 end
330 if msg_type == "Error" then
331 first_head = "\n" .. first_head
332 end
333 if string.sub(text, -1) ~= "\n" then
334 text = text .. " "
335 end
336 return first_head .. " "
337 .. string.gsub(
338 text
339 .. "on input line "

```

```

340 .. tex.inputlineno, "\n", "\n" .. cont .. " "
341)
342 .. "\n"
343 end

module_info Write messages.
module_warning local function module_info(mod, text)
module_error texio_write_nl("log", msg_format(mod, "Info", text))
346 end
347 luatexbase.module_info = module_info
348 local function module_warning(mod, text)
349 texio_write_nl("term and log",msg_format(mod, "Warning", text))
350 end
351 luatexbase.module_warning = module_warning
352 local function module_error(mod, text)
353 error(msg_format(mod, "Error", text))
354 end
355 luatexbase.module_error = module_error

(End of definition for module_info, module_warning, and module_error.)
Dedicated versions for the rest of the code here.

356 function luatexbase_warning(text)
357 module_warning("luatexbase", text)
358 end
359 function luatexbase_error(text)
360 module_error("luatexbase", text)
361 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.

```

362 local luaregisterbasetable = { }
363 local registermap = {
364 attributezero = "assign_attr" ,
365 charzero = "char_given" ,
366 CountZero = "assign_int" ,
367 dimenzero = "assign_dimen" ,
368 mathcharzero = "math_given" ,
369 muskipzero = "assign_mu_skip" ,
370 skipzero = "assign_skip" ,
371 tokszero = "assign_toks" ,
372 }
373 local createtoken
374 if tex.luatexversion > 81 then
375 createtoken = token.create
376 elseif tex.luatexversion > 79 then
377 createtoken = newtoken.create
378 end
379 local hashtokens = tex.hashtokens()
380 local luatexversion = tex.luatexversion
381 for i,j in pairs (registermap) do
382 if luatexversion < 80 then

```

```

383 luaregisterbasetable[hashtokens[i][1]] =
384 hashtokens[i][2]
385 else
386 luaregisterbasetable[j] = createtoken(i).mode
387 end
388 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.

```

389 local registernumber
390 if luatexversion < 80 then
391 function registernumber(name)
392 local nt = hashtokens[name]
393 if(nt and luaregisterbasetable[nt[1]]) then
394 return nt[2] - luaregisterbasetable[nt[1]]
395 else
396 return false
397 end
398 end
399 else
400 function registernumber(name)
401 local nt = createtoken(name)
402 if(luaregisterbasetable[nt.cmdname]) then
403 return nt.mode - luaregisterbasetable[nt.cmdname]
404 else
405 return false
406 end
407 end
408 end
409 luatexbase.registernumber = registernumber

```

(End of definition for `registernumber`.)

### 5.13 Attribute allocation

- new\_attribute** As attributes are used for Lua manipulations its useful to be able to assign from this end.

```

410 local attributes=setmetatable(
411 {}, {
412 __index = function(t,key)
413 return registernumber(key) or nil
414 end}
415)
416)
417 luatexbase.attributes = attributes
418 local attribute_count_name =
419 attribute_count_name or "e@alloc@attribute@count"
420 local function new_attribute(name)
421 tex_setcount("global", attribute_count_name,
422 tex_count[attribute_count_name] + 1)
423 if tex_count[attribute_count_name] > 65534 then
424 luatexbase_error("No room for a new \\attribute")

```

```

425 end
426 attributes[name] = tex_count[attribute_count_name]
427 luatexbase_log("Lua-only attribute " .. name .. " = " ..
428 tex_count[attribute_count_name])
429 return tex_count[attribute_count_name]
430 end
431 luatexbase.new_attribute = new_attribute

```

(End of definition for `new_attribute`.)

## 5.14 Custom whatsit allocation

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

```

432 local whatsit_count_name = whatsit_count_name or "e@alloc@whatsit@count"
433 local function new_whatsit(name)
434 tex_setcount("global", whatsit_count_name,
435 tex_count[whatsit_count_name] + 1)
436 if tex_count[whatsit_count_name] > 65534 then
437 luatexbase_error("No room for a new custom whatsit")
438 end
439 luatexbase_log("Custom whatsit " .. (name or "") .. " = " ..
440 tex_count[whatsit_count_name])
441 return tex_count[whatsit_count_name]
442 end
443 luatexbase.new_whatsit = new_whatsit

```

(End of definition for `new_whatsit`.)

## 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.

```

444 local bytecode_count_name =
445 bytecode_count_name or "e@alloc@bytecode@count"
446 local function new_bytecode(name)
447 tex_setcount("global", bytecode_count_name,
448 tex_count[bytecode_count_name] + 1)
449 if tex_count[bytecode_count_name] > 65534 then
450 luatexbase_error("No room for a new bytecode register")
451 end
452 luatexbase_log("Lua bytecode " .. (name or "") .. " = " ..
453 tex_count[bytecode_count_name])
454 return tex_count[bytecode_count_name]
455 end
456 luatexbase.new_bytecode = new_bytecode

```

(End of definition for `new_bytecode`.)

## 5.16 Lua chunk name allocation

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

```

457 local chunkname_count_name =
458 chunkname_count_name or "e@alloc@luachunk@count"
459 local function new_chunkname(name)

```

```

460 tex_setcount("global", chunkname_count_name,
461 tex_count[chunkname_count_name] + 1)
462 local chunkname_count = tex_count[chunkname_count_name]
463 chunkname_count = chunkname_count + 1
464 if chunkname_count > 65534 then
465 luatexbase_error("No room for a new chunkname")
466 end
467 lua.name[chunkname_count]=name
468 luatexbase_log("Lua chunkname " .. (name or "") .. " = " ..
469 chunkname_count .. "\n")
470 return chunkname_count
471 end
472 luatexbase.new_chunkname = new_chunkname

```

(End of definition for `new_chunkname`.)

## 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.

```

473 local luafunction_count_name =
474 luafunction_count_name or "e@alloc@luafunction@count"
475 local function new_luafunction(name)
476 tex_setcount("global", luafunction_count_name,
477 math.max(
478 #(lua.get_functions_table()),
479 tex_count[luafunction_count_name])
480 + 1)
481 lua.get_functions_table()[tex_count[luafunction_count_name]] = false
482 if tex_count[luafunction_count_name] > 65534 then
483 luatexbase_error("No room for a new luafunction register")
484 end
485 luatexbase_log("Lua function " .. (name or "") .. " = " ..
486 tex_count[luafunction_count_name])
487 return tex_count[luafunction_count_name]
488 end
489 luatexbase.new_luafunction = new_luafunction

```

(End of definition for `new_luafunction`.)

## 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.

Actually there are two tables: `realcallbacklist` directly contains the entries as described above while `callbacklist` only directly contains the already sorted entries. Other entries can be queried through `callbacklist` too which triggers a resort.

Additionally `callbackrules` describes the ordering constraints: It contains two element tables with the descriptions of the constrained callback implementations. It can additionally contain a `type` entry indicating the kind of rule. A missing value indicates a normal ordering constraint.

```

490 local realcallbacklist = {}
491 local callbackrules = {}
492 local callbacklist = setmetatable({}, {
493 __index = function(t, name)
494 local list = realcallbacklist[name]
495 local rules = callbackrules[name]
496 if list and rules then
497 local meta = {}
498 for i, entry in ipairs(list) do
499 local t = {value = entry, count = 0, pos = i}
500 meta[entry.description], list[i] = t, t
501 end
502 local count = #list
503 local pos = count
504 for i, rule in ipairs(rules) do
505 local rule = rules[i]
506 local pre, post = meta[rule[1]], meta[rule[2]]
507 if pre and post then
508 if rule.type then
509 if not rule.hidden then
510 assert(rule.type == 'incompatible-warning' and luatexbase_warning
511 or rule.type == 'incompatible-error' and luatexbase_error)(
512 "Incompatible functions \".. rule[1] .. \" and \".. rule[2]
513 .. \" specified for callback \".. name .. \".")
514 rule.hidden = true
515 end
516 else
517 local post_count = post.count
518 post.count = post_count+1
519 if post_count == 0 then
520 local post_pos = post.pos
521 if post_pos ~= pos then
522 local new_post_pos = list[pos]
523 new_post_pos.pos = post_pos
524 list[post_pos] = new_post_pos
525 end
526 list[pos] = nil
527 pos = pos - 1
528 end
529 pre[#pre+1] = post
530 end
531 end
532 end
533 for i=1, count do -- The actual sort begins
534 local current = list[i]
535 if current then

```

```

536 meta[current.value.description] = nil
537 for j, cur in ipairs(current) do
538 local count = cur.count
539 if count == 1 then
540 pos = pos + 1
541 list[pos] = cur
542 else
543 cur.count = count - 1
544 end
545 end
546 list[i] = current.value
547 else
548 -- Cycle occurred. TODO: Show cycle for debugging
549 -- list[i] = ...
550 local remaining = {}
551 for name, entry in next, meta do
552 local value = entry.value
553 list[#list + 1] = entry.value
554 remaining[#remaining + 1] = name
555 end
556 table.sort(remaining)
557 local first_name = remaining[1]
558 for j, name in ipairs(remaining) do
559 local entry = meta[name]
560 list[i + j - 1] = entry.value
561 for _, post_entry in ipairs(entry) do
562 local post_name = post_entry.value.description
563 if not remaining[post_name] then
564 remaining[post_name] = name
565 end
566 end
567 end
568 local cycle = {first_name}
569 local index = 1
570 local last_name = first_name
571 repeat
572 cycle[last_name] = index
573 last_name = remaining[last_name]
574 index = index + 1
575 cycle[index] = last_name
576 until cycle[last_name]
577 local length = index - cycle[last_name] + 1
578 table.move(cycle, cycle[last_name], index, 1)
579 for i=2, length//2 do
580 cycle[i], cycle[length + 1 - i] = cycle[length + 1 - i], cycle[i]
581 end
582 error('Cycle occurred at ' .. table.concat(cycle, ' -> ', 1, length))
583 end
584 end
585 end
586 realcallbacklist[name] = list
587 t[name] = list
588 return list
589 end

```

```
590 })
```

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

```
591 local list, data, exclusive, simple, reverselist = 1, 2, 3, 4, 5
592 local types = {
593 list = list,
594 data = data,
595 exclusive = exclusive,
596 simple = simple,
597 reverselist = reverselist,
598 }
```

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.)

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

Section 8.2: file discovery callbacks.

```
600 find_read_file = exclusive,
601 find_write_file = exclusive,
602 find_font_file = data,
603 find_output_file = data,
604 find_format_file = data,
605 find_vf_file = data,
606 find_map_file = data,
607 find_enc_file = data,
608 find_pk_file = data,
609 find_data_file = data,
610 find_opentype_file = data,
611 find_truetype_file = data,
612 find_type1_file = data,
613 find_image_file = data,

614 open_read_file = exclusive,
615 read_font_file = exclusive,
616 read_vf_file = exclusive,
617 read_map_file = exclusive,
618 read_enc_file = exclusive,
619 read_pk_file = exclusive,
620 read_data_file = exclusive,
621 read_truetype_file = exclusive,
622 read_type1_file = exclusive,
623 read_opentype_file = exclusive,
```

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

```
624 find_cidmap_file = data,
625 read_cidmap_file = exclusive,
```

Section 8.3: data processing callbacks.

```
626 process_input_buffer = data,
627 process_output_buffer = data,
628 process_jobname = data,
```

Section 8.4: node list processing callbacks.

```
629 contribute_filter = simple,
630 buildpage_filter = simple,
631 build_page_insert = exclusive,
632 pre_linebreak_filter = list,
633 linebreak_filter = exclusive,
634 append_to_vlist_filter = exclusive,
635 post_linebreak_filter = reverselist,
636 hpack_filter = list,
637 vpack_filter = list,
638 hpack_quality = exclusive,
639 vpack_quality = exclusive,
640 pre_output_filter = list,
641 process_rule = exclusive,
642 hyphenate = simple,
643 ligaturing = simple,
644 kerning = simple,
645 insert_local_par = simple,
646 % mlist_to_hlist = exclusive,
647 new_graf = exclusive,
```

Section 8.5: information reporting callbacks.

```
648 pre_dump = simple,
649 start_run = simple,
650 stop_run = simple,
651 start_page_number = simple,
652 stop_page_number = simple,
653 show_error_hook = simple,
654 show_warning_message = simple,
655 show_error_message = simple,
656 show_lua_error_hook = simple,
657 start_file = simple,
658 stop_file = simple,
659 call_edit = simple,
660 finish_synctex = simple,
661 wrapup_run = simple,
```

Section 8.6: PDF-related callbacks.

```
662 finish_pdffile = data,
663 finish_pdfpage = data,
664 page_objnum_provider = data,
665 page_order_index = data,
666 process_pdf_image_content = data,
```

Section 8.7: font-related callbacks.

```
667 define_font = exclusive,
668 glyph_info = exclusive,
669 glyph_not_found = exclusive,
```

```

670 glyph_stream_provider = exclusive,
671 make_extensible = exclusive,
672 font_descriptor_objnum_provider = exclusive,
673 input_level_string = exclusive,
674 provide_charproc_data = exclusive,
675 }
676 luatexbase.callbacktypes=callbacktypes

```

Sometimes multiple callbacks correspond to a single underlying engine level callback. Then the engine level callback should be registered as long as at least one of these callbacks is in use. This is implemented through a shared table which counts how many of the involved callbacks are currently in use. The engine level callback is registered iff this count is not 0.

We add `mlist_to_hlist` directly to the list to demonstrate this, but the handler gets added later when it is actually defined.

All callbacks in this list are treated as user defined callbacks.

```

677 local shared_callbacks = {
678 mlist_to_hlist = {
679 callback = "mlist_to_hlist",
680 count = 0,
681 handler = nil,
682 },
683 }
684 shared_callbacks.pre_mlist_to_hlist_filter = shared_callbacks.mlist_to_hlist
685 shared_callbacks.post_mlist_to_hlist_filter = shared_callbacks.mlist_to_hlist

```

`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.

```

686 local callback_register = callback_register or callback.register
687 function callback.register()
688 luatexbase_error("Attempt to use callback.register() directly\n")
689 end

```

(End of definition for `callback.register`.)

### 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.

```
690 local function data_handler(name)
691 return function(data, ...)
692 for _,i in ipairs(callbacklist[name]) do
693 data = i.func(data,...)
694 end
695 return data
696 end
697 end
```

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

```
698 local function data_handler_default(value)
699 return value
700 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.

```
701 local function exclusive_handler(name)
702 return function(...)
703 return callbacklist[name][1].func(...)
704 end
705 end
```

Handler for **list** callbacks.

```
706 local function list_handler(name)
707 return function(head, ...)
708 local ret
709 for _,i in ipairs(callbacklist[name]) do
710 ret = i.func(head, ...)
711 if ret == false then
712 luatexbase_warning(
713 "Function '" .. i.description .. "' returned false\n"
714 .. "in callback '" .. name .. "'"
715)
716 return false
717 end
718 if ret ~= true then
```

```

719 head = ret
720 end
721 end
722 return head
723 end
724 end

```

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

```

725 local function list_handler_default(head)
726 return head
727 end

```

Handler for `reverselist` callbacks.

```

728 local function reverselist_handler(name)
729 return function(head, ...)
730 local ret
731 local callbacks = callbacklist[name]
732 for i = #callbacks, 1, -1 do
733 local cb = callbacks[i]
734 ret = cb.func(head, ...)
735 if ret == false then
736 luatexbase_warning(
737 "Function '" .. cb.description .. "' returned false\n"
738 .. "in callback '" .. name .. "'")
739)
740 return false
741 end
742 if ret ~= true then
743 head = ret
744 end
745 end
746 return head
747 end
748 end

```

Handler for `simple` callbacks.

```

749 local function simple_handler(name)
750 return function(...)
751 for _,i in ipairs(callbacklist[name]) do
752 i.func(...)
753 end
754 end
755 end

```

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

```

756 local function simple_handler_default()
757 end

```

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

```

758 local handlers = {
759 [data] = data_handler,
760 [exclusive] = exclusive_handler,
761 [list] = list_handler,
762 [reverselist] = reverselist_handler,
763 [simple] = simple_handler,

```

```

764 }
765 local defaults = {
766 [data] = data_handler_default,
767 [exclusive] = nil,
768 [list] = list_handler_default,
769 [reverselist] = list_handler_default,
770 [simple] = simple_handler_default,
771 }
```

### 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.

```
772 local user_callbacks_defaults = {}
```

`create_callback` The allocator itself.

```

773 local function create_callback(name, ctype, default)
774 local ctype_id = types[ctype]
775 if not name or name == ""
776 or not ctype_id
777 then
778 luatexbase_error("Unable to create callback:\n" ..
779 "valid callback name and type required")
780 end
781 if callbacktypes[name] then
782 luatexbase_error("Unable to create callback '" .. name ..
783 "' :\ncallback is already defined")
784 end
785 default = default or defaults[ctype_id]
786 if not default then
787 luatexbase_error("Unable to create callback '" .. name ..
788 "' :\ndefault is required for '" .. ctype ..
789 "' , callbacks")
790 elseif type (default) ~= "function" then
791 luatexbase_error("Unable to create callback '" .. name ..
792 "' :\ndefault is not a function")
793 end
794 user_callbacks_defaults[name] = default
795 callbacktypes[name] = ctype_id
796 end
797 luatexbase.create_callback = create_callback
```

(End of definition for `create_callback`.)

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

```

798 local function call_callback(name,...)
799 if not name or name == "" then
800 luatexbase_error("Unable to create callback:\n" ..
801 "valid callback name required")
802 end
803 if user_callbacks_defaults[name] == nil then
804 luatexbase_error("Unable to call callback '" .. name
805 .. "' :\nunknown or empty")
```

```

806 end
807 local l = callbacklist[name]
808 local f
809 if not l then
810 f = user_callbacks_defaults[name]
811 else
812 f = handlers[callbacktypes[name]](name)
813 end
814 return f(...)
815 end
816 luatexbase.call_callback=call_callback

```

(End of definition for `call_callback`.)

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

```

817 local function add_to_callback(name, func, description)
818 if not name or name == "" then
819 luatexbase_error("Unable to register callback:\n" ..
820 "valid callback name required")
821 end
822 if not callbacktypes[name] or
823 type(func) ~= "function" or
824 not description or
825 description == "" then
826 luatexbase_error(
827 "Unable to register callback.\n\n"
828 .. "Correct usage:\n"
829 .. "add_to_callback(<callback>, <function>, <description>)"
830)
831 end

```

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

```

832 local l = realcallbacklist[name]
833 if l == nil then
834 l = { }
835 realcallbacklist[name] = l

```

Handle count for shared engine callbacks.

```

836 local shared = shared_callbacks[name]
837 if shared then
838 shared.count = shared.count + 1
839 if shared.count == 1 then
840 callback_register(shared.callback, shared.handler)
841 end

```

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

```

842 elseif user_callbacks_defaults[name] == nil then
843 callback_register(name, handlers[callbacktypes[name]](name))
844 end
845 end

```

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

```

846 local f = {
847 func = func,

```

```

848 description = description,
849 }
850 if callbacktypes[name] == exclusive then
851 if #l == 1 then
852 luatexbase_error(
853 "Cannot add second callback to exclusive function\n" ..
854 name .. '')
855 end
856 end
857 table.insert(l, f)
858 callbacklist[name] = nil

```

Keep user informed.

```

859 luatexbase_log(
860 "Inserting '" .. description .. "' in '" .. name .. "'."
861)
862 end
863 luatexbase.add_to_callback = add_to_callback

```

*(End of definition for add\_to\_callback.)*

**declare\_callback\_rule** Add an ordering constraint between two callback implementations

```

864 local function declare_callback_rule(name, desc1, relation, desc2)
865 if not callbacktypes[name] or
866 not desc1 or not desc2 or
867 desc1 == "" or desc2 == "" then
868 luatexbase_error(
869 "Unable to create ordering constraint. "
870 .. "Correct usage:\n"
871 .. "declare_callback_rule(<callback>, <description_a>, <description_b>)"
872)
873 end
874 if relation == 'before' then
875 relation = nil
876 elseif relation == 'after' then
877 desc2, desc1 = desc1, desc2
878 relation = nil
879 elseif relation == 'incompatible-warning' or relation == 'incompatible-error' then
880 elseif relation == 'unrelated' then
881 else
882 luatexbase_error(
883 "Unknown relation type in declare_callback_rule"
884)
885 end
886 callbacklist[name] = nil
887 local rules = callbackrules[name]
888 if rules then
889 for i, rule in ipairs(rules) do
890 if rule[1] == desc1 and rule[2] == desc2 or rule[1] == desc2 and rule[2] == desc1 then
891 if relation == 'unrelated' then
892 table.remove(rules, i)
893 else
894 rule[1], rule[2], rule.type = desc1, desc2, relation
895 end
896 return

```

```

897 end
898 end
899 if relation ~= 'unrelated' then
900 rules[#rules + 1] = {desc1, desc2, type = relation}
901 end
902 elseif relation ~= 'unrelated' then
903 callbackrules[name] = {{desc1, desc2, type = relation}}
904 end
905 end
906 luatexbase.declare_callback_rule = declare_callback_rule
(End of definition for declare_callback_rule.)

```

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

```

907 local function remove_from_callback(name, description)
908 if not name or name == "" then
909 luatexbase_error("Unable to remove function from callback:\n" ..
910 "valid callback name required")
911 end
912 if not callbacktypes[name] or
913 not description or
914 description == "" then
915 luatexbase_error(
916 "Unable to remove function from callback.\n\n"
917 .. "Correct usage:\n"
918 .. "remove_from_callback(<callback>, <description>)"
919)
920 end
921 local l = realcallbacklist[name]
922 if not l then
923 luatexbase_error(
924 "No callback list for '" .. name .. "'\n")
925 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.

```

926 local index = false
927 for i,j in ipairs(l) do
928 if j.description == description then
929 index = i
930 break
931 end
932 end
933 if not index then
934 luatexbase_error(
935 "No callback '" .. description .. "' registered for '" ..
936 name .. "'\n")
937 end
938 local cb = l[index]
939 table.remove(l, index)
940 luatexbase_log(
941 "Removing '" .. description .. "' from '" .. name .. "'."
942)
943 if #l == 0 then
944 realcallbacklist[name] = nil

```

```

945 callbacklist[name] = nil
946 local shared = shared_callbacks[name]
947 if shared then
948 shared.count = shared.count - 1
949 if shared.count == 0 then
950 callback_register(shared.callback, nil)
951 end
952 elseif user_callbacks_defaults[name] == nil then
953 callback_register(name, nil)
954 end
955 end
956 return cb.func, cb.description
957 end
958 luatexbase.remove_from_callback = remove_from_callback

```

(End of definition for `remove_from_callback`.)

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

```

959 local function in_callback(name, description)
960 if not name
961 or name == ""
962 or not realcallbacklist[name]
963 or not callbacktypes[name]
964 or not description then
965 return false
966 end
967 for _, i in pairs(realecallbacklist[name]) do
968 if i.description == description then
969 return true
970 end
971 end
972 return false
973 end
974 luatexbase.in_callback = in_callback

```

(End of definition for `in_callback`.)

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

```

975 local function disable_callback(name)
976 if(realecallbacklist[name] == nil) then
977 callback_register(name, false)
978 else
979 luatexbase_error("Callback list for " .. name .. " not empty")
980 end
981 end
982 luatexbase.disable_callback = disable_callback

```

(End of definition for `disable_callback`.)

`callback_descriptions` List the descriptions of functions registered for the given callback. This will sort the list if necessary.

```

983 local function callback_descriptions (name)
984 local d = {}
985 if not name
986 or name == ""

```

```

987 or not realcallbacklist[name]
988 or not callbacktypes[name]
989 then
990 return d
991 else
992 for k, i in pairs(callbacklist[name]) do
993 d[k]= i.description
994 end
995 end
996 return d
997 end
998 luatexbase.callback_descriptions =callback_descriptions

```

(End of definition for `callback_descriptions`.)

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

```

999 local function uninstall()
1000 module_info(
1001 "luatexbase",
1002 "Uninstalling kernel luatexbase code"
1003)
1004 callback.register = callback_register
1005 luatexbase = nil
1006 end
1007 luatexbase.uninstall = uninstall

```

(End of definition for `uninstall`.)

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

```

1008 create_callback('pre_mlist_to_hlist_filter', 'list')
1009 create_callback('mlist_to_hlist', 'exclusive', node.mlist_to_hlist)
1010 create_callback('post_mlist_to_hlist_filter', 'reverselist')
1011 function shared_callbacks.mlist_to_hlist.handler(head, display_type, need_penalties)
1012 local current = call_callback("pre_mlist_to_hlist_filter", head, display_type, need_penalties)
1013 if current == false then
1014 flush_list(head)
1015 return nil
1016 end
1017 current = call_callback("mlist_to_hlist", current, display_type, need_penalties)
1018 local post = call_callback("post_mlist_to_hlist_filter", current, display_type, need_penalties)
1019 if post == false then
1020 flush_list(current)
1021 return nil
1022 end
1023 return post
1024 end

```

(End of definition for `mlist_to_hlist`.)

1025 `</lua>`

Reset the catcode of @.

1026 `\tex\catcode`@=\etacatcode\relax`

# File 05

## ltxexpl.dtx

### 1 expl3-dependent code

#### 1.1 Loader

\@kernel@after@enddocument  
\@kernel@after@enddocument@afterlastpage

These two kernel hooks are used by the shipout code. They are defined earlier here because the lhooks code adds material to them.

```
1 {*2ekernel | latexrelease}
2 <{latexrelease}>\IncludeInRelease{2020/10/01}%
3 <{latexrelease}> {kernel@enddocument hooks}{Define several kernel hooks}
```

We only initialize these kernel hooks if they are not already existing. Otherwise they would be set to \empty on rollback which would be wrong because code that has been added to them may still have to be executed in the rollback situation. Instead code that writes to them needs to handle the rollback as needed. It is likely that we have to change that approach in the future, but for now it should do. (It is enough to test only for the existence of one hook, as all got added at the same time.)

```
4 \ifx\@kernel@after@enddocument\@undefined
5 \let\@kernel@after@enddocument\empty
6 \let\@kernel@after@enddocument@afterlastpage\empty
```

For the similar reasons we also define those that are used in \document because they too get material added to in early modules.

```
7 \let\@kernel@before@begindocument\empty
8 \let\@kernel@after@begindocument\empty
9 \fi
10 <{latexrelease}>\EndIncludeInRelease
11 <{latexrelease}>\IncludeInRelease{0000/00/00}%
12 <{latexrelease}> {kernel@enddocument hooks}{Define several kernel hooks}
13 <{latexrelease}>\let\@kernel@after@enddocument\@undefined
14 <{latexrelease}>\let\@kernel@after@enddocument@afterlastpage\@undefined
15 <{latexrelease}>\let\@kernel@before@begindocument\@undefined
16 <{latexrelease}>\let\@kernel@after@begindocument\@undefined
17 <{/2ekernel | latexrelease}>
18 <{latexrelease}>\EndIncludeInRelease
```

(End of definition for \@kernel@after@enddocument and others.)

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

```
19 <{*2ekernel | latexrelease}>
20 <{latexrelease}>\IncludeInRelease{2020/10/01}%
21 <{latexrelease}> {\@expl@sys@load@backend@@}{Roll forward support}%
22 \def\reserved@a{\ifdefined#1\else\def#1{}\fi}
23 \reserved@a\@expl@sys@load@backend@@
24 \reserved@a\@expl@push@filename@@
25 \reserved@a\@expl@push@filename@aux@@
26 \reserved@a\@expl@pop@filename@@
27 <{latexrelease}>\EndIncludeInRelease
28 <{/2ekernel | latexrelease}>
```

Create a hook for last-minute expl3 material.

```
29 {*2ekernel}
30 \def\@expl@finalise@setup@@{}
31 (/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.

```
32 {*2ekernel}
33 \long\def\@gobble#1{}
34 \long\def\@firstofone#1{#1}
35 \long\def\@firstoftwo#1#2{#1}
36 \long\def\@secondoftwo#1#2{#2}
37 \long\def\IfFileExists#1{%
38 \openin\@inputcheck"#1" %
39 \ifeof\@inputcheck
40 \expandafter\@secondoftwo
41 \else
42 \closein\@inputcheck
43 \expandafter\@firstoftwo
44 \fi}
45 \long\def\@ifnextchar#1#2#3{%
46 \let\reserved@d=#1%
47 \def\reserved@a{#2}%
48 \def\reserved@b{#3}%
49 \futurelet\@let@token\@ifnch}
50 \def\@ifnch{%
51 \ifx\@let@token\reserved@d
52 \expandafter\reserved@a
53 \else
54 \expandafter\reserved@b
55 \fi}
56 (/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!

```
57 {*2ekernel | latexrelease}
58 (<latexrelease>) \IncludeInRelease{2020/10/01}%
59 (<latexrelease> {expl3}{Pre-load expl3}%
60 \expandafter\ifx\csname tex\string _let:D\endcsname\relax
61 \expandafter\@firstofone
62 \else
63 \GenericInfo{}{Skipping: expl3 code already part of the format}%
64 <2ekernel> \expandafter\endinput
65 <latexrelease> \expandafter\@gobble
66 \fi
```

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

```
67 {%
68 \IfFileExists{expl3.ltx}%
69 {%
70 \ifnum0%
71 \ifdefined\pdffilesize 1\fi
72 \ifdefined\filesize 1\fi
73 \ifdefined\luatexversion\ifnum\luatexversion>94 1\fi\fi
74 }
```

```

74 \ifdefined\kanjiskip 1\fi
75 >0 %
76 \expandafter\@firstofone
77 \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 terminal in nonstop modes)
```

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.

```

78 <2ekernel> \def~{ }\def\MessageBreak{^^J~~~~~}%
79 <2ekernel> \errmessage{LaTeX Error:
80 <latexrelease> \@latex@error{%
81 LaTeX requires the e-TeX primitives and additional\MessageBreak
82 functionality available in the engines:\MessageBreak
83 - pdfTeX v1.40\MessageBreak
84 - XeTeX v0.99992\MessageBreak
85 - LuaTeX v0.95\MessageBreak
86 - e-(u)pTeX mid-2012\MessageBreak
87 or later%
88 <|latexrelease> } \@ehd \expandafter\@gobble
89 <2ekernel> } \batchmode \read -1 to \reserved@a
90 \fi
91 }
92 {%
93 <*2ekernel>
94 \errmessage{LaTeX requires expl3}%
95 \batchmode \read -1 to \reserved@a
96 </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.

```

97 <*latexrelease>
98 \@latex@warning@no@line
99 {You need a format that already contains a recent\MessageBreak
100 expl3 as part of the kernel, e.g. at least a kernel\MessageBreak
101 from 2019 to roll forward to that date!\MessageBreak
102 --- I'm giving up!\MessageBreak\MessageBreak
103 Note that manually loading the expl3 package\MessageBreak
104 from your distribution is not enough}%
105 \batchmode \read -1 to \reserved@a
106 </|latexrelease>
107 }
108 { \input expl3.ltx }%
109 }
110 <|latexrelease> \EndIncludeInRelease
111 <|latexrelease>

```

To support roll-forward for the case where `xparse` is fully integrated into the kernel, we do not need to repeat the complex test above as we can simply look for the marker command.

```

112 <|latexrelease> \IncludeInRelease{2020/02/02}%
113 <|latexrelease> {expl3}{Pre-load expl3}%

```

```

114 〈latexrelease〉\IfFileExists{expl3.ltx}{%
115 〈latexrelease〉 {%
116 〈latexrelease〉 \ifnum0%
117 〈latexrelease〉 \ifdefinable\pdffilesize 1\fi
118 〈latexrelease〉 \ifdefinable\filesize 1\fi
119 〈latexrelease〉 \ifdefinable\luatexversion\ifnum\luatexversion>94 1\fi\fi
120 〈latexrelease〉 >0 %
121 〈latexrelease〉 \else
122 〈latexrelease〉 \message{Skipping expl3-dependent extensions}
123 〈latexrelease〉 \expandafter\@gobbletwo
124 〈latexrelease〉 \fi
125 〈latexrelease〉 }
126 〈latexrelease〉 {%
127 〈latexrelease〉 \message{Skipping expl3-dependent extensions}%
128 〈latexrelease〉 \@gobbletwo
129 〈latexrelease〉 }%
130 〈latexrelease〉\input{expl3.ltx}
131 〈latexrelease〉\EndIncludeInRelease

```

Now in `\textrm{latexrelease}` mode, redefine a few commands to avoid “already defined” errors.

```
132 〈latexrelease〉\@ifundefined{ExplSyntaxOff}{}{\textrm{latexrelease}\@postltexp1}
```

## 1.2 Using `expl3` code

In order to ease the implementation of some new features in L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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.

```

133 〈latexrelease〉\IncludeInRelease{2020/10/01}{\@expl@cs@to@str@@N}%
134 〈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.

```

135 \ExplSyntaxOn
136 \cs_gset_eq:NN \@expl@cs@to@str@@N \cs_to_str:N
137 \cs_gset_eq:NN \@expl@str@if@eq@nnTF \str_if_eq:nnTF

```

```

138 \cs_gset_eq:NN \Expl@cs@prefix@spec@@N \cs_prefix_spec:N
139 \cs_if_exist:NTF \cs_parameter_spec:N
140 { \cs_gset_eq:NN \Expl@cs@parameter@spec@@N \cs_parameter_spec:N }
141 { \cs_gset_eq:NN \Expl@cs@parameter@spec@@N \cs_argument_spec:N }
142 \cs_gset_eq:NN __kernel_cs_parameter_spec:N \Expl@cs@parameter@spec@@N
143 \cs_gset_eq:NN \Expl@cs@replacement@spec@@N \cs_replacement_spec:N

144 \cs_gset_eq:NN \Expl@str@map@function@@NN \str_map_function:NN
145 \cs_gset_eq:NN \Expl@char@generate@@nn \char_generate:nn
146 \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.

```

147 \textrm{(latexrelease)}\EndIncludeInRelease
148 \textrm{(latexrelease)}\IncludeInRelease{0000/00/00}{\Expl@cs@to@str@@N}%
149 \textrm{(latexrelease)} \Expl3 macros added for the 2020-10-01 release}%
150 \textrm{(latexrelease)}\let \Expl@cs@to@str@@N \undefined
151 \textrm{(latexrelease)}\let \Expl@str@if@eq@nnTF \undefined
152 \textrm{(latexrelease)}\let \Expl@cs@prefix@spec@@N \undefined
153 \textrm{(latexrelease)}\let \Expl@cs@parameter@spec@@N \undefined
154 \textrm{(latexrelease)}\let \Expl@cs@replacement@spec@@N \undefined
155 \textrm{(latexrelease)}\let \Expl@str@map@function@@NN \undefined
156 \textrm{(latexrelease)}\EndIncludeInRelease
157 \textrm{/2ekernel | latexrelease)}

```

## 2 Document-level command names for `expl3` functions

Current home for L3 programming layer functions that we make directly available at the document level. This section may need to be moved later (after `\NewDocumentCommand` is defined in case we want to use that in the setup).

`\fpeval` The expandable command `\fpeval` takes as its argument a floating point expression and produces a result using the normal rules of mathematics. As this command is expandable it can be used where TeX requires a number and for example within a low-level `\edef` operation to give a purely numerical result. See `usrguide3` for further explanation.

`\inteval` The expandable command `\inteval` takes as its argument an integer expression and `\dimeval` produces a result using the normal rules of mathematics. The operations recognised are `\skipeval` `+, -, *` and `/` plus parentheses. Division occurs with *rounding*, and ties are rounded away from zero. As this command is expandable it can be used where TeX requires a number and for example within a low-level `\edef` operation to give a purely numerical result. See `usrguide3` for further explanation. `\dimeval` and `\skipeval` are similar, but generate fixed and rubber length values, respectively.

`\fpeval` A document level wrapper around the code level function for floating point calculations.  
`\inteval`  
`\dimeval`  
`\skipeval`

```

158 \textrm{(*2ekernel | latexrelease)}
159 \textrm{(latexrelease)}\IncludeInRelease{2022/06/01}%
160 \textrm{(latexrelease)} \fpeval{fp and int calculations}%
161 \ExplSyntaxOn
162 \cs_new_eq:NN \fpeval \fp_eval:n

```

And a few more, this time wrappers around the eTeX primitives.

```
163 \cs_new_eq:NN \inteval \int_eval:n
164 \cs_new_eq:NN \dimeval \dim_eval:n
165 \cs_new_eq:NN \skipEval \skip_eval:n
166 \ExplSyntaxOff

(End of definition for \fpeval and others.)
167 ⟨/2ekernel | latexrelease⟩
168 ⟨latexrelease⟩\EndIncludeInRelease
169 ⟨latexrelease⟩\IncludeInRelease{0000/00/00} %
170 ⟨latexrelease⟩ \fpeval}{fp and int calculations} %
171 ⟨latexrelease⟩
172 ⟨latexrelease⟩\let\fpeval@undefined
173 ⟨latexrelease⟩\let\inteval@undefined
174 ⟨latexrelease⟩\let\dimeval@undefined
175 ⟨latexrelease⟩\let\skipEval@undefined
176 ⟨latexrelease⟩\EndIncludeInRelease
```

**\UserName** When declaring new commands with `\NewDocumentCommand` or `\NewCommandCopy`  
**\ExpandArgs** or similar, it is sometimes necessary to “construct” the csname. As a general mechanism  
the L3 programming layer has `\exp_args:N...` for this, but there is no mechanism for  
it if `\ExplSyntaxOn` is not active. We therefore offer a few of these commands also with  
CamelCase names.

**\UserName** A document wrapper for changing arguments to cs names for use with `\NewDocumentCommand`  
**\ExpandArgs** and similar functions.

```
177 ⟨*2ekernel | latexrelease⟩
178 ⟨latexrelease⟩\IncludeInRelease{2022/06/01} %
179 ⟨latexrelease⟩ \ExpandArgs}{Some pre-expansion commands} %
180 \ExplSyntaxOn
181 \cs_new_eq:NN \UserName \use:c

182 \cs_new:Npn \ExpandArgs #1
183 {
184 \cs_if_exist_use:cF { \exp_args:N #1 }
185 { \msg_expandable_error:nnn { kernel } { unknown-arg-expansion } {#1} }
186 }
187 \msg_new:nnn { kernel } { unknown-arg-expansion }
188 { Unknown-arg-expansion~"#1" }
189 \ExplSyntaxOff
```

(End of definition for \UserName and \ExpandArgs.)

```
190 ⟨/2ekernel | latexrelease⟩
191 ⟨latexrelease⟩\EndIncludeInRelease
192 ⟨latexrelease⟩\IncludeInRelease{0000/00/00} %
193 ⟨latexrelease⟩ \ExpandArgs}{Some pre-expansion commands} %
194 ⟨latexrelease⟩
195 ⟨latexrelease⟩\let\UserName@undefined
196 ⟨latexrelease⟩\let\ExpandArgs@undefined
197 ⟨latexrelease⟩\EndIncludeInRelease
```

**\IfExplAtLeastTF** A pretty simple wrapper.

```
\IfExplAtLeastTF
198 {*2ekernel | latexrelease}
199 \IncludeInRelease{2023/11/01}%
200 \IfExplAtLeastTF{Test for expl3 date}%
201 \def\IfExplAtLeastTF{\ifl@t@r\ExplLoaderFileDate}
```

(End of definition for `\IfExplAtLeastTF`.)

We make sure the command is always available.

```
202 /2ekernel | latexrelease
203 \EndIncludeInRelease
204 \IncludeInRelease{0000/00/00}%
205 \IfExplAtLeastTF{Test for expl3 date}%
206 \IfExplAtLeastTF{
207 \def\IfExplAtLeastTF{\ifl@t@r\ExplLoaderFileDate}
208 \EndIncludeInRelease
```

# File 06

## ltdefns.dtx

### 1 Definitions

This section contains commands used in defining other macros.

1 `(*2ekernel)`

#### 1.1 Initex initializations

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

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

(*End of definition for \two@digits.*)

`\typeout` Display something on the terminal.

3 `</2ekernel>`  
4 `<*2ekernel | latexrelease>`  
5 `<latexrelease>\IncludeInRelease{2020/10/01}%`  
6 `<latexrelease> \typeout{\allow "par" in \typeout}%`  
7 `\protected\long\def\typeout#1{\begingroup`  
8  `\set@display@protect`  
9  `\def\par{\^J}%`  
10  `\immediate\write\@unused{#1}\endgroup}`  
11 `</2ekernel | latexrelease>`  
12 `<latexrelease>\EndIncludeInRelease`  
13 `<latexrelease>\IncludeInRelease{0000/00/00}%`  
14 `<latexrelease> \typeout{\allow "par" in \typeout}%`  
15 `<latexrelease>`  
16 `<latexrelease>\def\typeout#1{\begingroup\set@display@protect`  
17  `\immediate\write\@unused{#1}\endgroup}`  
18 `<latexrelease>\EndIncludeInRelease`  
19 `(*2ekernel)`

(*End of definition for \typeout.*)

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

20 `\newlinechar`\^J`

(*End of definition for \newlinechar.*)

#### 1.2 Saved versions of TeX primitives

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

`\@@par`

21 `\let\@@par=\par`  
22 `%\let\@@input=\input %% moved earlier`  
23 `%\let\@@end=\end %%`

(*End of definition for \@@par.*)

\@@hyph Save original primitive definition.  
<sup>24</sup> \let\@@hyph=\-  
*(End of definition for \@@hyph.)*

\@@italiccorr Save the original italic correction.  
<sup>25</sup> \let\@@italiccorr=\/  
*(End of definition for \@@italiccorr.)*

\@height The following definitions save token space. E.g., using \@height instead of height saves  
\@depth 5 tokens at the cost in time of one macro expansion.  
\@width <sup>26</sup> \def\@height{height} \def\@depth{depth} \def\@width{width}  
\@minus <sup>27</sup> \def\@minus{minus}  
\@plus <sup>28</sup> \def\@plus{plus}

The next one is another 100 tokens worth.  
<sup>29</sup> \def\hb@xt@{\hbox to}  
*(End of definition for \@height and others.)*

<sup>30</sup> \message{hacks,}  
\hb@xt@

### 1.3 Command definitions

This section defines the following commands:

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

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

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

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

\@dblarg {\langle CMD\rangle}{\langle ARG\rangle}  
Expands to \{\langle CMD\rangle\}[\langle ARG\rangle]\{\langle ARG\rangle\}. Use \@dblarg\CS when \CS takes arguments [ARG1]{ARG2}, where default is ARG1 = ARG2.

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

\@ifdefinable {\NAME}{\langle YES\rangle} Executes \langle YES\rangle 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 \*{\langle FOO\rangle}{\langle i\rangle}{\langle TEXT\rangle}  
User command to define \FOO to be a macro with i arguments (i = 0 if missing) having the definition \langle TEXT\rangle. Produces an error if \FOO already defined.

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

\renewcommand \*{\langle FOO\rangle}{\langle i\rangle}{\langle TEXT\rangle}

```

Same as \newcommand, except it checks if \FOO already defined.
\newenvironment *{\{FOO\}}[\langle i\rangle]{\{DEF1\}}{\{DEF2\}}
equivalent to:
\newcommand{\FOO}[i]{\def{\endFOO}{DEF2}}
(or the appropriate star forms).

\renewenvironment
 Obvious companion to \newenvironment.

\@cons : See description of \output routine.
\@car \@car T1 T2 ... Tn\@nil == T1 (unexpanded)
\@cdr \@cdr T1 T2 ... Tn\@nil == T2 ... Tn (unexpanded)
\typeout {\langle message\rangle}
Produces a warning message on the terminal.

\typein {\langle message\rangle}
Types message, asks the user to type in a command, then executes it
\typein [(\CS)]{\MSG}
Same as above, except defines \CS to be the input instead of executing it.

\typein
 31 \def\typein{%
 32 \let\@typein\relax
 33 \@testopt\@xtypein\@typein}

 34 \ifx\directlua\undefined
 35 \def\@xtypein[#1]#2{%
 36 \typeout{#2}%
 37 \advance\endlinechar\@M
 38 \read\@inputcheck to#1%
 39 \advance\endlinechar-\@M
 40 \@typein}%

 41 \else
 42 \def\@xtypein[#1]#2{%
 43 \typeout{#2}%
 44 \begingroup \endlinechar\m@ne
 45 \read\@inputcheck to#1%
 46 \expandafter\endgroup
 47 \expandafter\def\expandafter#1\expandafter{#1}%
 48 \@typein}%

 49 \fi
(End of definition for \typein.)

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

(End of definition for \@namedef.)

\@nameuse
 51 \def\@nameuse#1{\csname #1\endcsname}
(End of definition for \@nameuse.)

```

```

\@cons
52 \def\@cons#1#2{\begingroup\let\@elt\relax\xdef#1{#1\@elt #2}\endgroup}
(End of definition for \@cons.)
```

```

\@car
\@cdr 53 \def\@car#1#2\@nil{#1}
54 \def\@cdr#1#2\@nil{#2}
(End of definition for \@car and \@cdr.)
```

```

\@carcube \@carcube T1 ... Tn\@nil = T1 T2 T3 , n > 3
55 ⟨/2ekernel⟩
56 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@carcube}{Make \@carcube long}%
57 ⟨*2ekernel | latexrelease⟩
58 \long\def\@carcube#1#2#3#4\@nil{#1#2#3}
59 ⟨/2ekernel | latexrelease⟩
60 ⟨latexrelease⟩\EndIncludeInRelease
61 %
62 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@carcube}{Undo: Make \@carcube long}%
63 ⟨latexrelease⟩\def\@carcube#1#2#3#4\@nil{#1#2#3}
64 ⟨latexrelease⟩\EndIncludeInRelease
65 ⟨*2ekernel⟩
(End of definition for \@carcube.)
```

\@onlypreamble 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.

```

66 \def\preamblecmds{}
67 \def\@onlypreamble#1{%
68 \expandafter\gdef\expandafter\preamblecmds\expandafter{%
69 \preamblecmds\do#1}}
70 \@onlypreamble\@onlypreamble
71 \@onlypreamble\preamblecmds
```

(End of definition for \@onlypreamble and \preamblecmds.)

\@star@or@long Look ahead for a \*. If present reset \l@ngrel@x so that the next definition, #1, will be non-long.

```

72 \def\@star@or@long#1{%
73 \@ifstar
74 {\let\l@ngrel@x\relax#1}%
75 {\let\l@ngrel@x\long#1}}
```

(End of definition for \@star@or@long.)

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

```

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

(End of definition for \l@ngrel@x.)

\newcommand User level \newcommand.

```

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

```

\new@command 78 \def\new@command#1{%
79 \@testopt{\@newcommand#1}0}

(End of definition for \newcommand and \new@command.)

```

\@newcommand Handling arguments for \newcommand.

```

\@argdef 80 \def\@newcommand#1[#2]{%
81 \kernel@ifnextchar [{\@xargdef#1[#2]}%
82 {\@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.

```

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

```

Handle the second optional argument.

```

85 \long\def\@xargdef#1[#2][#3]#4{%
86 \@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.

```

87 \expandafter\def\expandafter#1\expandafter{%
88 \expandafter
89 \@protected@testopt
90 \expandafter
91 #1%
92 \csname\string#1\endcsname
93 {#3}}%

```

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

```

94 \expandafter\@yargdef
95 \csname\string#1\endcsname
96 \tw@
97 {#2}%
98 {#4}}}

```

(End of definition for \@newcommand, \@argdef, and \@xargdef.)

\@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.

```

99 \long\def\@testopt#1#2{%
100 \kernel@ifnextchar[{\#1}{#1[{#2}]}}}

```

(End of definition for \@testopt.)

\@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 \cx\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.

```

101 \def\@protected@testopt#1{%
102 \ifx\protect\@typeset@protect
103 \expandafter\@testopt
104 \else
105 \c{x}\protect#1%
106 \fi}

```

(End of definition for \@protected@testopt.)

\@yargdef These generate a primitive argument specification, from a L<sup>A</sup>T<sub>E</sub>X [*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 L<sup>A</sup>T<sub>E</sub>X2.09 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.

```

107 \long \def \@yargdef #1#2#3{%
108 \ifx#2\tw@
109 \def\reserved@b##1{####1}%
110 \else
111 \let\reserved@b\@gobble
112 \fi
113 \expandafter
114 \@yargd@f \expandafter{\number #3}#1%
115 }

116 \long \def \@yargd@f#1#2{%
117 \def \reserved@a ##1##2##{%
118 \expandafter\def\expandafter#2\reserved@b ##1##
119 }%
120 \l@ngrel@x \reserved@a 0##1##2##3##4##5##6##7##8##9##1%
121 }

```

(End of definition for \@yargdef and \@yargd@f.)

\@reargdef

```

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

```

(End of definition for \@reargdef.)

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

```
125 \def\renew@command#1{%
126   \begingroup \escapechar`m@ne\xdef\@gtempa{{\string#1}}\endgroup
127   \expandafter\@ifundefined\@gtempa
128     {\@latex@error{Command \string#1 undefined}\@ehc}%
129     \relax
130   \let\@ifdefinable\@rc@ifdefinable
131   \new@command#1}
```

(End of definition for \renewcommand and \renew@command.)

- \@ifdefinable Test if user is allowed to define a command.
- \@@ifdefinable \long\def\@ifdefinable #1#2{%
133 \edef\reserved@a{\expandafter\@gobble\string #1}%
134 \@ifundefined\reserved@a
135 {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
136 \ifx \reserved@b\@qend \@notdefinable\else
137 \ifx \reserved@a\@qrelax \@notdefinable\else
138 #2%
139 \fi
140 \fi}%
141 \@notdefinable}
- Saved definition of \@ifdefinable.
- ```
142 \let\@@ifdefinable\@ifdefinable
```
- Version of \@ifdefinable for use with \renewcommand. Does not do the check this time, but restores the normal definition.
- ```
143 \long\def\@rc@ifdefinable#1#2{%
144   \let\@ifdefinable\@@ifdefinable
145   #2}
```

(End of definition for \@ifdefinable, \@@ifdefinable, and \@rc@ifdefinable.)

- \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.

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

- ```
147 \def\newenvironment#1{%
148 \@testopt{\@newenva#1}0}
```

```

149 \def\@newenva#1[#2]{%
150 \kernel@ifnextchar [{{\@newenvb#1[#2]}{\@newenv{#1}{[#2]}}}}
151 \def\@newenvb#1[#2][#3]{\@newenv{#1}{[#2]}{[#3]}}
152 \def\renewenvironment{\@star@or@long\renew@environment}
153 \def\renew@environment#1{%
154 \@ifundefined{#1}{%
155 {\@latex@error{Environment #1 undefined}\@ehc
156 }\relax
157 \expandafter\let\csname#1\endcsname\relax
158 \expandafter\let\csname end#1\endcsname\relax
159 \new@environment{#1}}
160 \long\def\@newenv#1#2#3#4{%
161 \@ifundefined{#1}{%
162 {\expandafter\let\csname#1\expandafter\endcsname
163 \csname end#1\endcsname}%
164 }\relax
165 \expandafter\new@command
166 {\csname #1\endcsname#2{#3}%
167 \l@ngrel@x\expandafter\def\csname end#1\endcsname{#4}}
168 \def\newif#1{%
169 \count@\escapechar \escapechar\m@ne
170 \let#1\iffalse
171 \@if#1\iftrue
172 \@if#1\iffalse
173 \escapechar\count@}

```

(End of definition for `\newenvironment` and `\renewenvironment`.)

`\@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 by temporarily letting the undefined `\...` (begin code) to the definition of `\end...` and as a result we get an error if that has a definition.

(End of definition for `\@newenv`.)

`\newif` And here’s a different sort of allocation: For example, `\newif\iffloor` creates `\foottrue`, `\foofalse` to go with `\iffloor`.

```

168 \def\newif#1{%
169 \count@\escapechar \escapechar\m@ne
170 \let#1\iffalse
171 \@if#1\iftrue
172 \@if#1\iffalse
173 \escapechar\count@}

```

```

\@if 174 \def\@if#1#2{%
175 \expandafter\def\csname\expandafter\@gobbletwo\string#1%
176 \expandafter\@gobbletwo\string#2\endcsname
177 {\let#1#2}}

```

(End of definition for `\newif` and `\@if`.)

`\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.

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

```

\provide@command 179 \def\provide@command#1{%
180 \begingroup
181 \escapechar\m@ne\xdef\@gtempa{\string#1}%
182 \endgroup
183 \expandafter\@ifundefined\@gtempa
184 {\def\reserved@a{\new@command#1}%
185 {\def\reserved@a{\renew@command\reserved@a}%
186 \reserved@a}%

```

(End of definition for `\providecommand` and `\provide@command`.)

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

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

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

```
188 \onlypreamble\CheckCommand
```

```

\check@command 189 \def\check@command#1#2{\@check@c#1{#2}}
190 \onlypreamble\check@command

```

(End of definition for `\CheckCommand` and `\check@command`.)

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

```

191 \long\def\@check@c#1#2#3{%
192 \expandafter\let\csname\string\reserved@a\endcsname\relax
193 \renew@command\reserved@a#2{#3}%
194 \@ifundefined{\string\reserved@a}%
195 {\@check@eq#1\reserved@a}%

```

```

196 {\expandafter\@check@eq
197 \csname\string#1\expandafter\endcsname
198 \csname\string\reserved@a\endcsname}}
199 \onlypreamble\@check@c

(End of definition for \@check@c.)
```

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

```

200 \def\@check@eq#1#2{%
201 \ifx#1#2\else
202 \@latex@warning@no@line
203 {Command \noexpand#1 has
204 changed.\MessageBreak
205 Check if current package is valid}%
206 \fi}
207 \onlypreamble\@check@eq

(End of definition for \@check@eq.)
```

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

```

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

(End of definition for \@gobble and others.)
```

\@firstofone Some argument-grabbers.

```

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

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

(End of definition for \@firstofone and others.)
```

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

```

216 \long\def\@thirddofthree#1#2#3{#3}

(End of definition for \@thirddofthree.)
```

\@expandtwoargs A macro to totally expand two arguments to another macro

```

217 </2ekernel>
218 <latexrelease>\IncludeInRelease{2022/11/01}%
219 <latexrelease> {\@expandtwoargs}{protected edef}%
220 <*2ekernel | latexrelease>
221 \def\@expandtwoargs#1#2#3{%
222 \protected@edef\reserved@a{\noexpand#1{#2}{#3}}\reserved@a}
223 </2ekernel | latexrelease>
224 <latexrelease>\EndIncludeInRelease
225 <latexrelease>\IncludeInRelease{00/00/00}%
226 <latexrelease> {\@expandtwoargs}{protected edef}%
227 <latexrelease>\def\@expandtwoargs#1#2#3{%
228 \def\@expandtwoargs#1#2#3{\noexpand#1{#2}{#3}}\reserved@a}
229 <latexrelease>\EndIncludeInRelease
230 <*2ekernel>
```

(End of definition for \@expandtwoargs.)

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

(End of definition for \@backslashchar.)

## 1.4 Robust commands and protect

Fragile and robust commands are one of the thornier issues in L<sup>A</sup>T<sub>E</sub>X's commands. Whilst typesetting documents, L<sup>A</sup>T<sub>E</sub>X makes use of many of T<sub>E</sub>X's features, such as arithmetic, defining macros, and setting variables. However, there are (at least) three different occasions when these commands are not safe. These are called 'moving arguments' by L<sup>A</sup>T<sub>E</sub>X, and consist of:

- writing information to a file, such as indexes or tables of contents.
- writing information to the screen.
- inside an \edef, \message, \mark, or other command which evaluates its argument fully.

The method L<sup>A</sup>T<sub>E</sub>X 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.
- \cunexpandable@protect, for writing a moving argument to a file. So \protect\foo will write \protect\foo followed by a space. This value is also used inside \edefs, \marks and other commands which evaluate their arguments fully. More precisely, whenever the content of an \edef or \xdef etc. can contain arbitrary user input not under the direct control of the programmer, one should use \protected@edef instead of \edef, etc., so that \protect has a suitable definition and the user input will not break if it contains fragile commands.

\cunexpandable@protect  
232 \def\cunexpandable@protect{\noexpand\protect\noexpand}  
(End of definition for \cunexpandable@protect.)

\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 `\haligns`. If you really have to do such a thing, then please set `\@typeset@protect` to be `\empty` as well. (This is what the code for `\patterns` does, for example.)

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

```
233 \def\DeclareRobustCommand{\@star@or@long\declare@robustcommand}
234 \def\declare@robustcommand#1{
235 \ifx#1\undefined\else\ifx#1\relax\else
236 \@latex@info{Redefining \string#1}
237 \fi\fi
238 \edef\reserved@a{\string#1}
239 \def\reserved@b{\#1}
240 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}
241 \edef#1{
242 \ifx\reserved@a\reserved@b
243 \noexpand\x@protect
244 \noexpand#1
245 \fi
246 \noexpand\protect
247 \expandafter\noexpand\csname
248 \expandafter\gobble\string#1 \endcsname
249 }%
250 \let\@ifdefinable\@rc@ifdefinable
251 \expandafter\new@command\csname
252 \expandafter\gobble\string#1 \endcsname
253 }
```

(End of definition for `\DeclareRobustCommand` and `\declare@robustcommand`.)

```
\x@protect
254 \def\x@protect#1{
255 \ifx\protect\@typeset@protect\else
256 \x@protect#1
257 \fi
258 }
259 \def\x@protect#1\fi#2#3{
260 \fi\protect#1
261 }
```

(End of definition for `\x@protect` and `\x@protect`.)

\@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.

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

(End of definition for \@typeset@protect.)

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

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

(End of definition for \set@display@protect and \set@typeset@protect.)

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

```
265 \def\protected@edef{%
266 \let\@@protect\protect
267 \let\protect\@unexpandable@protect
268 \afterassignment\restore@protect
269 \edef
270 }
271 \def\protected@xdef{%
272 \let\@@protect\protect
273 \let\protect\@unexpandable@protect
274 \afterassignment\restore@protect
275 \xdef
276 }
277 \def\unrestored@protected@xdef{%
278 \let\protect\@unexpandable@protect
279 \xdef
280 }
281 \def\restore@protect{\let\protect\@@protect}
```

(End of definition for \protected@edef and others.)

\protect The normal meaning of \protect

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

(End of definition for \protect.)

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

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

```
283 </2ekernel>
```

```
284 <latexrelease>\IncludeInRelease{2020/10/01}{\MakeRobust}{\MakeRobust}%
```

```

285 {*2ekernel | latexrelease}
286 \def\MakeRobust#1{%
287 \count@=\escapechar
288 \escapechar='\\
289 \@ifundefined{\expandafter\gobble\string#1}{%
290 \if@error{Command '\string#1' undefined.%}
291 \MessageBreak There is nothing here to make robust}%
292 \eha
293 }%

```

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.

```

294 {%
295 \ifundefined{\expandafter\gobble\string#1\space}{%
296 {%
297 \expandafter\@kernel@rename@newcommand
298 \csname\expandafter\gobble\string#1\space\endcsname
299 #1%
300 \edef\reserved@a{\string#1}%
301 \def\reserved@b{#1}%
302 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
303 \xdef#1{%
304 \ifx\reserved@a\reserved@b
305 \noexpand\x@protect\noexpand#1%
306 \fi
307 \noexpand\protect\expandafter\noexpand
308 \csname\expandafter\gobble\string#1\space\endcsname}%
309 }%
310 {\@latex@info{Command '\string#1' is already robust}}%
311 }%
312 \escapechar=\count@
313 }%

```

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.

```

314 \def\@kernel@rename@newcommand#1#2{%
315 \robust@command@chk@safe#2%
316 {\@if@newcommand#2%
317 {\afterassignment\global
318 \global\copy@newcommand#1#2%
319 \global\let#2@\undefined

```

```

320 \global\expandafter\let\csname string#2\endcsname\@undefined}%
321 {\global\let#1=#2}%%
322 {\global\let#1=#2}

323 (/2ekernel | latexrelease)
324 \end{IncludeInRelease}
325 %
326 \end{IncludeInRelease}[2019/10/01]{\MakeRobust}{\MakeRobust}%
327 \def\MakeRobust#1{%
328 \IfUndefined{\expandafter\gobble\string#1}{%
329 \IfLatexError{The control sequence '\string#1' is undefined!}%
330 {MessageBreak There is nothing here to make robust}%
331 \ch@%
332 }%
333 }%
334 \IfUndefined{\expandafter\gobble\string#1\space}{%
335 }%
336 \global\expandafter\let\csname
337 \expandafter\gobble\string#1\space\endcsname=\#1%
338 \edef\reserved@a{\string#1}%
339 \def\reserved@b{\#1}%
340 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
341 \xdef#1{%
342 \ifx\reserved@a\reserved@b
343 \noexpand\x@protect\noexpand#1%
344 \fi
345 \noexpand\protect\expandafter\noexpand
346 \csname\expandafter\gobble\string#1\space\endcsname}%
347 }%
348 {\IfLatexInfo{The control sequence '\string#1' is already robust}}%
349 }%
350 }%
351 \let\@kernel@rename@newcommand\@undefined
352 \end{IncludeInRelease}
353 %
354 \end{IncludeInRelease}[2015/01/01]{\MakeRobust}{\MakeRobust}%
355 \def\MakeRobust#1{%
356 \IfUndefined{\expandafter\gobble\string#1}{%
357 \IfLatexError{The control sequence '\string#1' is undefined!}%
358 {MessageBreak There is nothing here to make robust}%
359 \ch@%
360 }%
361 }%
362 \IfUndefined{\expandafter\gobble\string#1\space}{%
363 }%
364 \expandafter\let\csname
365 \expandafter\gobble\string#1\space\endcsname=\#1%
366 \edef\reserved@a{\string#1}%
367 \def\reserved@b{\#1}%
368 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
369 \xdef#1{%
370 \ifx\reserved@a\reserved@b
371 \noexpand\x@protect\noexpand#1%
372 \fi

```

```

373 〈\latexrelease〉 \noexpand\protect\expandafter\noexpand
374 〈\latexrelease〉 \csname\expandafter@gobble\string#1\space\endcsname}%
375 〈\latexrelease〉 }%
376 〈\latexrelease〉 {\@latex@info{The control sequence ‘\string#1’ is already robust}}%
377 〈\latexrelease〉 }%
378 〈\latexrelease〉}%
379 〈\latexrelease〉\let\@kernel@rename@newcommand\@undefined
380 〈\latexrelease〉\EndIncludeInRelease
381 %
382 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\MakeRobust}{\MakeRobust}%
383 〈\latexrelease〉\let\MakeRobust\@undefined
384 〈\latexrelease〉\let\@kernel@rename@newcommand\@undefined
385 〈\latexrelease〉\EndIncludeInRelease
386 {*ekernel}

```

(End of definition for `\MakeRobust` and `\@kernel@rename@newcommand`.)

`\kernel@make@fragile` The opposite of `\MakeRobust` except that it doesn't do many checks as it is internal to the kernel. Why does one want such a thing? Only for compatibility reasons if `\latexrelease` requests a rollback of the kernel. For this reason we pretend that this command existed in all earlier versions of L<sup>A</sup>T<sub>E</sub>X 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.

```

387 〈/2ekernel〉
388 〈*2ekernel | \latexrelease〉
389 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
390 〈\latexrelease〉 {\@kernel@make@fragile}{Undo robustness}%
391 〈\def\kernel@make@fragile#1{%
392 \@ifundefined{\expandafter\@gobble\string#1\space}%

```

If not robust do nothing.

```
393 {}%
```

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_`.

```

394 {}%
395 \global\expandafter\let\expandafter #1\csname
396 \expandafter@gobble\string#1\space\endcsname
397 \expandafter\@kernel@rename@newcommand
398 \csname\expandafter\@gobble\string#1\expandafter\endcsname
399 \csname\expandafter\@gobble\string#1\space\endcsname
400 \global\expandafter\let\csname
401 \expandafter\@gobble\string#1\space\endcsname\@undefined
402 }%
403 }

404 〈\latexrelease〉\EndIncludeInRelease
405 %
406 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
407 〈\latexrelease〉 {\@kernel@make@fragile}{Undo robustness}%
408 〈\latexrelease〉\def\kernel@make@fragile#1{%
409 〈\latexrelease〉 \ifundefined{\expandafter\@gobble\string#1\space}%
410 〈\latexrelease〉 {}%
411 〈\latexrelease〉 {}%
```

```

412 〈\latexrelease〉 \global\expandafter\let\expandafter #1\csname
413 〈\latexrelease〉 \expandafter\@gobble\string#1\space\endcsname
414 〈\latexrelease〉 \global\expandafter\let\csname
415 〈\latexrelease〉 \expandafter\@gobble\string#1\space\endcsname\@undefined
416 〈\latexrelease〉 }%
417 〈\latexrelease〉}
418 〈\latexrelease〉\EndIncludeInRelease
419 〈/2ekernel | \latexrelease〉
420 〈*2ekernel〉

```

(End of definition for \kernel@make@fragile.)

## 1.5 Acting on robust commands

```

421 〈/2ekernel〉
422 〈\latexrelease〉\IncludeInRelease{2020-10-01}{\robust@command@act}
423 〈\latexrelease〉 {Add \robust@command@act}%
424 〈*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 <sub>$\varepsilon$</sub>  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 <sub>$\varepsilon$</sub>  kernel, like commands defined with xpars'e'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 \ShowCommand 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>}-{<false>}. 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.

```

425 \long\def\robust@command@act#1#2#3#4{%
426 \robust@command@chk@safe#2%
427 {\expandafter\robust@command@act@loop
428 \expandafter#2%
429 #1{\@nnil\@nnil}%
430 \robust@command@act@end}%
431 {\robust@command@act@end}%
432 {#3}{#4}}%

```

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.

```

433 \long\def\robust@command@act@loop#1#2{\robust@command@act@loop@aux#1#2}
434 \long\def\robust@command@act@loop@aux#1#2#3{%
435 \ifx\@nnil#2%
436 \else
437 #2{#1}%
438 {\robust@command@act@do{#3}}%
439 {\expandafter\robust@command@act@loop\expandafter#1}%
440 \fi}
441 \long\def\robust@command@act@do#1%
442 \fi#2%
443 \robust@command@act@end#3#4{%
444 \fi
445 #1#4}

```

If the end is reached and no action was taken, then do `<fallback-action><act-arg>`.

```
\robust@command@act@end 446 \long\def\robust@command@act@end#1#2{#1#2}
```

```

447 \long\def\robust@command@chk@safe#1{%
448 \begingroup
449 \escapechar='\\
450 \expandafter\endgroup\expandafter
451 \robust@command@act@chk@args\meaning#1:->\@nil}
452 \def\robust@command@act@chk@args#1:->#2\@nil{%
453 \cexpl@str@if@eq@@nnTF{#1}{macro}%
454 {\cfirstoftwo}%
455 {\cexpl@str@if@eq@@nnTF{#1}{\protected macro}%
456 {\cfirstoftwo}%
457 {\csecondoftwo}}}}

458 {/2ekernel | latexrelease}
459 (latexrelease)\EndIncludeInRelease
460 (latexrelease)\IncludeInRelease{0000-00-00}{\robust@command@act}
461 (latexrelease) {Add \robust@command@act}%

```

```

462 〈\latexrelease〉\let\robust@command@act\@undefined
463 〈\latexrelease〉\let\robust@command@act@loop\@undefined
464 〈\latexrelease〉\let\robust@command@act@loop@aux\@undefined
465 〈\latexrelease〉\let\robust@command@act@do\@undefined
466 〈\latexrelease〉\let\robust@command@act@end\@undefined
467 〈\latexrelease〉\let\robust@command@chk@safe\@undefined
468 〈\latexrelease〉\let\robust@command@act@chk@args\@undefined
469 〈\latexrelease〉\EndIncludeInRelease
470 {*2ekernel}

```

(End of definition for \robust@command@act and others.)

### 1.5.1 Copying robust commands

```

471 〈/2ekernel〉
472 〈\latexrelease〉\IncludeInRelease{2020-10-01}{\DeclareCommandCopy}
473 〈\latexrelease〉 {Add \NewCommandCopy, \RenewCommandCopy, and \DeclareCommandCopy}%
474 {*2ekernel | latexrelease}

```

\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:

```

\ProvideCommandCopy \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.

```

475 \def\NewCommandCopy{%
476 \declare@commandcopy
477 {\@firstofone}%
478 {\@firstoftwo\@notdefinable}}%
479 \def\RenewCommandCopy{%
480 \declare@commandcopy
481 {\@latex@error{Command \@backslashchar\reserved@a\space undefined}\@ehc
482 \@firstofone}%
483 {\@firstofone}}%
484 \def\DeclareCommandCopy{%
485 \declare@commandcopy
486 {\@firstofone}%
487 {\@firstofone}}

```

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>`.

```

488 \long\def\declare@commandcopy#1#2#3#4{%
489 \edef\reserved@a{\@expl@cs@to@str@@N#3}%
490 \@ifundefined\reserved@a{#1}{#2}%
491 {\declare@commandcopy@do{#3}{#4}}}
492 \long\def\declare@commandcopy@do#1#2{%
493 \robust@command@act
494 \@declarecommandcopylisthook#2%
495 \declare@commandcopy@let{#1#2}}

```

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

```

496 \def\@declarecommandcopylisthook{%
497 {\@if@DeclareRobustCommand \@copy@DeclareRobustCommand}%
498 {\@if@newcommand \@copy@newcommand}}

```

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

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

`\declare@commandcopy@let`

Now the rollback code.

```

500 {/2ekernel | latexrelease}
501 <latexrelease>\EndIncludeInRelease
502 <latexrelease>\IncludeInRelease{0000-00-00}{\DeclareCommandCopy}
503 <latexrelease> {Undefine \NewCommandCopy, \RenewCommandCopy, and \DeclareCommandCopy}%
504 <latexrelease>\let\NewCommandCopy\@undefined
505 <latexrelease>\let\RenewCommandCopy\@undefined
506 <latexrelease>\let\DeclareCommandCopy\@undefined
507 <latexrelease>\let\declare@commandcopy\@undefined
508 <latexrelease>\let@\declarecommandcopylisthook\@undefined
509 <latexrelease>\let\declare@commandcopy@let\@undefined
510 <latexrelease>\EndIncludeInRelease
511 {*2ekernel}

```

(End of definition for `\NewCommandCopy` and others.)

```

512 {/2ekernel}
513 <latexrelease>\IncludeInRelease{2023-06-01}{\DeclareEnvironmentCopy}
514 <latexrelease> {Add \NewEnvironmentCopy, \RenewEnvironmentCopy, and \DeclareEnvironmentCopy}%
515 {*2ekernel | latexrelease}

```

`\NewEnvironmentCopy` If `\#1` or `\end#1` already exist one gets an error message talking about the problematical command (not the environment). The remainder of the L<sup>A</sup>T<sub>E</sub>X run is probably badly broken and it is unlikely that continuing it gives reasonable results.

```

516 \def\NewEnvironmentCopy{%
517 \declare@environmentcopy
518 {\@firstofone}%
519 {\@firstoftwo\@notdefinable}%
520 \def\RenewEnvironmentCopy{%
521 \declare@environmentcopy
522 {\@latex@error{Environment \reserved@a\space undefined}\@ehc
523 {\@firstofone}%
524 {\@firstofone}}}

```

```

525 \def\DeclareEnvironmentCopy{%
526 \declare@environmentcopy
527 {\@firstofone}%
528 {\@firstofone}%
529 \long\def\declare@environmentcopy#1#2#3#4{%
530 \edef\reserved@a{\@ifundefined{#3}{\end#3}{#3}}%
531 \@ifundefined\reserved@a
532 {\def\reserved@a{#3}#1}%
533 {\def\reserved@a{#3}#2}%
534 {\ExpandArgs{cc}\declare@commandcopy@do{#3}{#4}}%
535 {\ExpandArgs{cc}\declare@commandcopy@do{\end#3}{\end#4}}}%

```

Now the rollback code.

```

536 </2ekernel | latexrelease>
537 <latexrelease>\EndIncludeInRelease
538 <latexrelease>\IncludeInRelease{0000-00-00}{\DeclareEnvironmentCopy}
539 <latexrelease> {\Undefine \NewEnvironmentCopy, \RenewEnvironmentCopy, and \DeclareEnvironmentCopy}
540 <latexrelease>\let\NewEnvironmentCopy\@undefined
541 <latexrelease>\let\RenewEnvironmentCopy\@undefined
542 <latexrelease>\let\DeclareEnvironmentCopy\@undefined
543 <latexrelease>\EndIncludeInRelease
544 <*2ekernel>

```

(*End of definition for \NewEnvironmentCopy , \RenewEnvironmentCopy , and \DeclareEnvironmentCopy .*)

### 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 is detected to have 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: we use \robust@command@act to iterate through the \@showcommandlisthook list, and if nothing is found, fallback to \show.

```

545 </2ekernel>
546 <latexrelease>\IncludeInRelease{2020-10-01}{\ShowCommand}%
547 <latexrelease> {\Add \ShowCommand}%
548 <*2ekernel | latexrelease>

549 \long\def\ShowCommand#1{%
550 \robust@command@act
551 \@showcommandlisthook#1%
552 \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 initialized to cope with \DeclareRobustCommand and \newcommand with optional arguments.

```

553 \def\@showcommandlisthook{%
554 {\@if@DeclareRobustCommand \@show@DeclareRobustCommand}%
555 {\@if@newcommand \@show@newcommand}}

```

Now the rollback code.

```

556 </2ekernel | latexrelease>
557 <latexrelease>\EndIncludeInRelease

```

```

558 〈\latexrelease〉\IncludeInRelease{0000-00-00}{\ShowCommand}
559 〈\latexrelease〉 {Undefined \ShowCommand}%
560 〈\latexrelease〉\let\ShowCommand\@undefined
561 〈\latexrelease〉\let\@showcommandlisthook\@undefined
562 〈\latexrelease〉\EndIncludeInRelease
563 {*2ekernel}

(End of definition for \ShowCommand and \@showcommandlisthook.)
```

```

564 〈/2ekernel〉
565 〈\latexrelease〉\IncludeInRelease{2020-10-01}{\@if@DeclareRobustCommand}
566 〈\latexrelease〉 {Add \@if@DeclareRobustCommand, \@if@newcommand,
567 〈\latexrelease〉 \@copy@DeclareRobustCommand, \@copy@newcommand,
568 〈\latexrelease〉 \@show@DeclareRobustCommand, \@show@newcommand}%
569 {*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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s robust types. These tests are heavily based on Heiko's `\LetLtxMacro`, but chopped into separate macros.

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

```

570 \long\def\@if@DeclareRobustCommand#1{%
571 \begingroup
572 \escapechar='\\
573 \edef\reserved@a{\string#1}%
574 \edef\reserved@b{\detokenize{#1}}%
575 \xdef\@gtempa{%
576 \ifx\reserved@a\reserved@b
577 \noexpand\x@protect
578 \noexpand#1%
579 \fi
580 \noexpand\protect
581 \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname}%
582 \endgroup
583 \ifx\@gtempa#1\relax
584 \expandafter\@firstoftwo
585 \else
586 \expandafter\@secondoftwo
587 \fi}

```

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.

```

588 \long\def\@copy@DeclareRobustCommand#1#2{%
589 \begingroup
590 \escapechar='\\

```

```

591 \edef\reserved@a{\string#1}%
592 \edef\reserved@b{\detokenize{#1}}%
593 \edef\reserved@a{%
594 \endgroup
595 \def\noexpand#1{%
596 \ifx\reserved@a\reserved@b
597 \noexpand\x@protect
598 \noexpand#1%
599 \fi
600 \noexpand\protect
601 \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname}%
602 \noexpand\copy@kernel@robust@command
603 \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname
604 \expandafter\noexpand\csname\@expl@cs@to@str@@N#2 \endcsname}%
605 \reserved@a}%
606 \long\def\copy@kernel@robust@command#1#2{%
607 \robust@command@chk@safe#2%
608 {\@if@newcommand#2%
609 {\@copy@newcommand}%
610 {\declare@commandcopy@let}%
611 {\declare@commandcopy@let}%
612 #1#2}

```

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`.

```

613 \long\def\@show@DeclareRobustCommand#1{%
614 \typeout{> \string#1=robust macro:}%
615 \typeout{>@\expl@cs@replacement@spec@@N#1.^~J}%
616 \expandafter\show@kernel@robust@command
617 \csname\@expl@cs@to@str@@N#1 \endcsname}%
618 \long\def\show@kernel@robust@command#1{%
619 \robust@command@chk@safe#1%
620 {\@if@newcommand#1%
621 {\@show@newcommand}%
622 {\@show@macro}%
623 {\@show@macro}%
624 #1}
625 \let\@show@macro\show

```

(End of definition for `\@if@DeclareRobustCommand` and others.)

### 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.

```

626 \long\def\@if@newcommand#1{%
627 \edef\reserved@a{%
628 \noexpand\@protected@testopt
629 \noexpand#1%
630 \expandafter\noexpand\csname\@backslashchar\expl@cs@to@str@@N#1\endcsname}%
631 \edef\reserved@b{%
632 \unexpanded\expandafter\expandafter\expandafter
633 {\expandafter\@car\@cube#1{}{}{\@nil}}}%
634 \ifx\reserved@a\reserved@b
635 \expandafter\@firstoftwo
636 \else
637 \expandafter\@secondoftwo
638 \fi}

```

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>}.

```

639 \long\def\@copy@newcommand#1#2{%
640 \edef#1{\noexpand\@protected@testopt
641 \noexpand#1%
642 \expandafter\noexpand\csname\@backslashchar\expl@cs@to@str@@N#1\endcsname
643 \unexpanded\expandafter\expandafter\expandafter
644 {\expandafter\@gobblethree#2}}%
645 \expandafter
646 \let\csname\@backslashchar\expl@cs@to@str@@N#1\expandafter\endcsname
647 \csname\@backslashchar\expl@cs@to@str@@N#2\endcsname}

```

\@show@newcommand 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.

```

648 \long\def\@show@newcommand#1{%
649 \typeout{> \string#1=robust macro:}%
650 \typeout{->\@expl@cs@replacement@spec@@N#1.^~J}%
651 \expandafter\@show@newcommand@aux
652 \csname\@backslashchar\expl@cs@to@str@@N#1\expandafter\endcsname
653 \expandafter{\#1}\@show@tokens}

```

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`.

```
654 \long\def\@show@newcommand@aux#1#2#3{%
655 \typeout{> \string#1=\@expl@cs@prefix@spec@@N#1macro:}%
656 #3{default \string##1=\expandafter\detokenize@gobblethree#2.^~J%
657 \@expl@cs@parameter@spec@@N#1->\@expl@cs@replacement@spec@@N#1}}
```

This macro prints the contents of the token list (macro) #1 using `\showtokens`. The `\expandafter` gymnastics ensures that `\showtokens` itself, and the internals of this macro aren't showed in the context lines.

```
658 \long\def\@show@tokens#1{%
659 \edef\reserved@a{\#1}%
660 \showtokens\expandafter
661 \expandafter\expandafter{\expandafter\reserved@a}}
```

Now the rollback code.

```
662 </2ekernel | latexrelease>
663 <latexrelease>\EndIncludeInRelease
664 <latexrelease>\IncludeInRelease{0000-00-00}{\@if@DeclareRobustCommand}
665 <latexrelease> {Undefine \@if@DeclareRobustCommand, \@if@newcommand,
666 <latexrelease> \copy@DeclareRobustCommand, \copy@newcommand,
667 <latexrelease> \show@DeclareRobustCommand, \show@newcommand}%
668 <latexrelease>\let\@if@DeclareRobustCommand\@undefined
669 <latexrelease>\let\copy@DeclareRobustCommand\@undefined
670 <latexrelease>\let\show@DeclareRobustCommand\@undefined
671 <latexrelease>\let\@if@newcommand\@undefined
672 <latexrelease>\let\copy@newcommand\@undefined
673 <latexrelease>\let\show@newcommand\@undefined
674 %
675 <latexrelease>\let\copy@kernel@robust@command\@undefined
676 <latexrelease>\let\show@kernel@robust@command\@undefined
677 <latexrelease>\let\show@newcommand@aux\@undefined
678 <latexrelease>\EndIncludeInRelease
679 <*2ekernel>
```

(End of definition for `\@if@newcommand` and others.)

### 1.5.5 Showing environments

`\ShowEnvironment`

```
680 </2ekernel>
681 <latexrelease>\IncludeInRelease{2023-06-01}{\ShowEnvironment}
682 <latexrelease> {Add \ShowEnvironment}%
683 <*2ekernel | latexrelease>
```

`\ShowEnvironment` is quite similar to `\ShowCommand`. We will pass the environment `(env)` around as the macro `\env`, because `\robust@command@act` expects a single token.

```
684 \def\ShowEnvironment#1{%
685 \expandafter\@show@environment\csname #1\endcsname}
686 \long\def\@show@environment#1{%
687 \robust@command@act}
```

```

688 \@showenvironmentlisthook#1%
689 \@show@normalenv#1}

```

This is similar to `\@showcommandlisthook`, but uses the dedicated versions for environments.

```

690 \def\@showenvironmentlisthook{%
691 {\@if@DeclareRobustCommand \@show@DeclareRobustCommand@env}%
692 {\@if@newcommand \@show@newcommand@env}}

```

These are similar to the command versions below, except they say “environment” and call `\@show@environment@end` to print the `\end` part.

```

693 \long\def\@show@newcommand@env#1{%
694 \@show@environment@begin#1%
695 \expandafter\@show@newcommand@aux
696 \csname\@backslashchar\@expl@cs@to@str@@N#1\expandafter\endcsname
697 \expandafter{\#1}\@show@typeout
698 \@show@environment@end#1}
699 \long\def\@show@DeclareRobustCommand@env#1{%
700 \@show@environment@begin#1%
701 \begingroup
702 \let\@show@tokens\@show@typeout
703 \let\@show@macro\@show@nonstop
704 \expandafter\show@kernel@robust@command
705 \csname\@expl@cs@to@str@@N#1 \endcsname
706 \endgroup
707 \@show@environment@end#1}
708 \long\def\@show@environment@begin#1{%
709 \typeout{> \string\begin{\@expl@cs@to@str@@N#1}=environment :}%
710 \typeout{\@expl@cs@parameter@spec@@N#1->%
711 \@expl@cs@replacement@spec@@N#1.^~J}}

```

A “normal” environment is straightforward. `\@show@environment@end` needs to check if the `\end` part is defined and show it accordingly, otherwise the output would show gibberish.

```

712 \long\def\@show@normalenv#1{%
713 \@show@environment@begin#1%
714 \@show@environment@end#1}
715 \long\def\@show@environment@end#1{%
716 \expandafter\@show@environment@end@aux
717 \csname end\@expl@cs@to@str@@N#1\endcsname#1}
718 \long\def\@show@environment@end@aux#1#2{%
719 \@show@tokens{\string\end{\@expl@cs@to@str@@N#2}%
720 \ifx\relax#1=undefined%
721 \else:^~J\@expl@cs@parameter@spec@@N#1->%
722 \@expl@cs@replacement@spec@@N#1%
723 \fi}}

```

And here some auxiliaries:

```

\@show@nonstop
\@show@typeout

```

`\@show@nonstop` same output as `\show`, but doesn’t stop for interaction;

`\@show@typeout` same output as `\showtokens`, but doesn’t stop for interaction.

```

724 \def\@show@nonstop#1{%
725 \typeout{> \string#1=\@expl@cs@prefix@spec@@N#1macro:^^J%
726 \@expl@cs@parameter@spec@@N#1->\@expl@cs@replacement@spec@@N#1.}}
727 \def\@show@typeout#1{\typeout{> #1.^^J}}

```

Now the rollback code.

```

728 {/2ekernel | latexrelease}
729 <latexrelease>\EndIncludeInRelease
730 <latexrelease>\IncludeInRelease{0000-00-00}{\ShowEnvironment}
731 <latexrelease> {Undefine \ShowEnvironment}%
732 <latexrelease>\let\ShowEnvironment\undefined
733 <latexrelease>\EndIncludeInRelease
734 {*2ekernel}

```

(End of definition for \ShowEnvironment and others.)

## 1.6 Internal defining commands

These commands are used internally to define other L<sup>A</sup>T<sub>E</sub>X commands.

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

```

735 {/2ekernel}
736 <latexrelease>\IncludeInRelease{2018-04-01}{\@ifundefined}
737 <latexrelease>{Leave commands undefined in \@ifundefined}%
738 {*2ekernel | latexrelease}

```

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

```

739 \def\@ifundefined#1{%
740 \ifcsname#1\endcsname\@ifundefined@d@i\else\@ifundefined@d@ii\fi{#1}}
741 \long\def\@ifundefined@d@i#1\fi#2{#1}
742 \expandafter\ifx\csname #2\endcsname\relax
743 \@ifundefined@d@ii
744 \fi
745 \@secondoftwo}
746 \long\def\@ifundefined@d@ii\fi#1#2#3{\fi #2}

```

Now test of engine.

```
747 \ifx\numexpr\undefined
```

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

```

748 \def\@ifundefined#1{%
749 \expandafter\ifx\csname#1\endcsname\relax
750 \expandafter\@firstoftwo
751 \else
752 \expandafter\@secondoftwo
753 \fi}
754 \else\ifx\directlua\undefined

```

Use the \ifcsname defined above.

```
755 \else
```

Optimised version for LuaTeX, using \lastnamedcs

```
756 \def\@ifundefined#1{%
757 \ifcsname#1\endcsname
758 \expandafter\ifx\lastnamedcs\relax\else\@ifundefined@#1\fi
759 \fi
760 \@firstoftwo}
761 \long\def\@ifundefined@#1#2#3#4#5{#1#2#5}
762 \fi
763 \fi
764 </2ekernel | latexrelease>
765 <latexrelease>\EndIncludeInRelease
766 <latexrelease>\IncludeInRelease{0000-00-00}{\@ifundefined}
767 <latexrelease>\{Leave commands undefined in \@ifundefined\}%
768 <latexrelease>\def\@ifundefined#1{%
769 <latexrelease> \expandafter\ifx\csname#1\endcsname\relax
770 <latexrelease> \expandafter\@firstoftwo
771 <latexrelease> \else
772 <latexrelease> \expandafter\@secondoftwo
773 <latexrelease> \fi}
774 <latexrelease>\EndIncludeInRelease
775 <*2ekernel>
```

(End of definition for \@ifundefined.)

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

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

(End of definition for \@qend and \@qrelax.)

\@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.

```
778 \long\def\@ifnextchar#1#2#3{%
779 \let\reserved@d=#1%
780 \def\reserved@a{#2}%
781 \def\reserved@b{#3}%
782 \futurelet\@let@token\@ifnch}
```

(End of definition for \@ifnextchar.)

\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.

```
783 \let\kernel@ifnextchar\@ifnextchar
```

(End of definition for \kernel@ifnextchar.)

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

```

784 \def\@ifnch{%
785 \ifx\@let@token\@sptoken
786 \let\reserved@c\@xifnch
787 \else
788 \ifx\@let@token\reserved@d
789 \let\reserved@c\reserved@a
790 \else
791 \let\reserved@c\reserved@b
792 \fi
793 \fi
794 }
```

(End of definition for \@ifnch.)

\@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.

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

(End of definition for \@sptoken.)

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

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

(End of definition for \@xifnch.)

\@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.

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

(End of definition for \@ifstar.)

```

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

(End of definition for \@dblarg and \@xdblarg.)

\@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.

```

800 \def\@sanitize{\@makeother\ \ @makeother\\@\makeother\$@\makeother\&%
801 \@\makeother\#\@\makeother\^\@\makeother_@\makeother\%\@\makeother\~}
```

(End of definition for \@sanitize.)

`\@onellevel@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.

```
802 \def \@onellevel@sanitize #1{%
803 \edef #1{\expandafter\strip@prefix
804 \meaning #1}%
805 }
```

(End of definition for `\@onellevel@sanitize`.)

`\string@makeletter` 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.

```
806 </2ekernel>
807 <latexrelease>\IncludeInRelease{2020/10/01}{\string@makeletter}
808 <latexrelease> {Add \string@makeletter}%
809 <*2ekernel | latexrelease>
810 \def\string@makeletter#1{%
811 \@expl@str@map@function@@NN#1\@string@makeletter}
812 \def\@string@makeletter#1{%
813 \char@if@alph{#1}%
814 {\@expl@char@generate@nn{'#1}{11}}%
815 {#1}}
816 \def\char@if@alph#1{%
817 \ifnum0\ifnum`#1<`A 1\fi\ifnum`#1>`z 1\fi
818 \if\ifnum`#1>`Z @\fi\ifnum`#1<`a @\fi01\fi>0
819 \expandafter\@secondoftwo
820 \else
821 \expandafter\@firstoftwo
822 \fi}
823 </2ekernel | latexrelease>
824 <latexrelease>\EndIncludeInRelease
825 %
826 <latexrelease>\IncludeInRelease{0000/00/00}{\string@makeletter}
827 <latexrelease> {Undefined \string@makeletter}%
828 <latexrelease>\let\string@makeletter\@undefined
829 <latexrelease>\let\@string@makeletter\@undefined
830 <latexrelease>\let\char@if@alph\@undefined
831 <latexrelease>\EndIncludeInRelease
832 <*2ekernel>
```

(End of definition for `\string@makeletter`, `\@string@makeletter`, and `\char@if@alph`.)

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

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

(End of definition for `\makeatletter` and `\makeatother`.)

## 2 Discretionary Hyphenation

\-  
\@dischyp 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 hyphenation.

L<sup>A</sup>T<sub>E</sub>X, from L<sup>A</sup>T<sub>E</sub>X2.09 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 T<sub>E</sub>X definition is saved as \@@hyph just in case anyone needs it.

L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub>, between 1993 and 2017, had a comment at this point saying that the definition "would probably change" because the definition always uses -. The definition used below was given in comments at this point during time.

In 2017 we finally enabled this definition by default, with the older L<sup>A</sup>T<sub>E</sub>X definition accessible via `latexrelease` as usual.

In LuaL<sup>A</sup>T<sub>E</sub>X 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 LuaL<sup>A</sup>T<sub>E</sub>X compatible with language specific hyphenation characters.

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

```
835 \def\@latex@info#1{%
836 \ifx\@kernel\@latexrelease
837 \IncludeInRelease{2020/10/01}{\-\{Use primitive \- in Lua\LaTeX\}\%}
838 \else
839 \ifx\directlua\undefined
840 \DeclareRobustCommand{\-}{%
841 \discretionary{-}{ }{ }
842 \char \ifnum\hyphenchar>\z@ \defaulthyphenchar
843 \else \hyphenchar\font
844 \fi }{}%
845 }{}%
846 }{}%
847 }{}%
848 }{}%
849 \else
850 \let\-\@\@hyph
851 \fi
852 \ifx\@kernel\@latexrelease
853 \EndIncludeInRelease
854 \else
855 \IncludeInRelease{2017/04/15}{\-\{Use \hyphenchar in \-}\%}
856 \DeclareRobustCommand{\-}{%
857 \discretionary{-}{ }{ }
858 \char \ifnum\hyphenchar>\z@ \defaulthyphenchar
859 \else }
```

```

860 〈\latexrelease〉 \hyphenchar\font
861 〈\latexrelease〉 \fi
862 〈\latexrelease〉 }{}{%
863 〈\latexrelease〉
864 〈\latexrelease〉\EndIncludeInRelease
865 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\-}{Use \hyphenchar in \-}%
866 〈\latexrelease〉\def\-\{\discretionary{-}{}{}\}
867 〈\latexrelease〉\EndIncludeInRelease

868 〈*2ekernel | \latexrelease〉
869 〈\let\@dischyp=\-
870 〈/2ekernel | \latexrelease〉
871 〈*2ekernel〉

(End of definition for \- and \@dischyp.)
Delayed from ltvers.dtx
872 \newif\if@includeinrelease
873 \if@includeinreleasefalse
Delayed from ltpplain.dtx
874 〈/2ekernel〉
875 〈*2ekernel | \latexrelease〉
876 〈\latexrelease〉\IncludeInRelease{2019/10/01}%
877 〈\latexrelease〉 {\allowbreak}{Make various commands robust}%
878 \MakeRobust\allowbreak
879 \MakeRobust\bigbreak
880 \MakeRobust\break
881 \MakeRobust\dotfill
882 \MakeRobust\frenchspacing
883 \MakeRobust\goodbreak
884 \MakeRobust\hrulefill
885 \MakeRobust\medbreak
886 \MakeRobust\nobreak
887 \MakeRobust\nonfrenchspacing
888 \MakeRobust\obeylelines
889 \MakeRobust\obeyspaces
890 \MakeRobust\slash
891 \MakeRobust\smallbreak
892 \MakeRobust\strut
893 \MakeRobust\underbar
894 〈/2ekernel | \latexrelease〉
895 〈\latexrelease〉\EndIncludeInRelease
896 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
897 〈\latexrelease〉 {\allowbreak}{Make various commands robust}%
898 〈\latexrelease〉
899 〈\latexrelease〉\kernel@make@fragile\allowbreak
900 〈\latexrelease〉\kernel@make@fragile\bigbreak
901 〈\latexrelease〉\kernel@make@fragile\break
902 〈\latexrelease〉\kernel@make@fragile\dotfill
903 〈\latexrelease〉\kernel@make@fragile\frenchspacing
904 〈\latexrelease〉\kernel@make@fragile\goodbreak
905 〈\latexrelease〉\kernel@make@fragile\hrulefill
906 〈\latexrelease〉\kernel@make@fragile\medbreak
907 〈\latexrelease〉\kernel@make@fragile\nobreak
908 〈\latexrelease〉\kernel@make@fragile\nonfrenchspacing

```

```
909 〈latexrelease〉\kernel@make@fragile\obeylines
910 〈latexrelease〉\kernel@make@fragile\obeyspaces
911 〈latexrelease〉\kernel@make@fragile\slash
912 〈latexrelease〉\kernel@make@fragile\smallbreak
913 〈latexrelease〉\kernel@make@fragile\strut
914 〈latexrelease〉\kernel@make@fragile\underbar
915 〈latexrelease〉
916 〈latexrelease〉\EndIncludeInRelease
917 〈*2ekernel〉
```

\g@addto@macro Globally add to the end of a macro. This macro is used by the kernel to add to its internal hooks.

```
918 \long\def\g@addto@macro#1#2{%
919 \begingroup
920 \toks@\expandafter{#1#2}%
921 \xdef#1{\the\toks@}%
922 \endgroup}
```

(*End of definition for \g@addto@macro.*)

```
923 〈/2ekernel〉
```

# File 07

## ltcmd.dtx

### 1 Creating document commands

Document commands should be created using the tools provided by this module: `\NewDocumentCommand`, etc., in almost all cases. This allows clean separation of document-level syntax from code-level interfaces. Users have a need to create new document commands, and as such a significant amount of documentation for `ltcmd` is provided as part of `usrguide3`. Here, additional material aimed at programmers is provided

```
1 (@@=cmd)
2 {*2ekernel}
3 \message{document commands,}
4 {/2ekernel}
```

`ltcmd` code contains an `\^@` character, which usually has catcode 15, so `\IncludeInRelease` will break when this code is being skipped, so we'll save the catcode of `\^@` to restore later:

```
5 {*2ekernel | latexrelease}
6 {latexrelease}\edef\@latexrelease@catcode@null{\the\catcode'\^@ }
7 {latexrelease}\catcode'\^@=12
8 \ExplSyntaxOn
9 {latexrelease}\NewModuleRelease{2020/10/01}{ltcmd}
10 {latexrelease} {Document~command~parser}%
```

#### 1.1 Variables and constants

---

`\l__cmd_arg_spec_tl` Holds the argument specification after normalization of shorthands.

```
11 \tl_new:N \l__cmd_arg_spec_tl
```

---

`\l__cmd_args_tl` Token list variable for grabbed arguments.

```
12 \tl_new:N \l__cmd_args_tl
```

---

`\l__cmd_args_i_tl` Hold the modified arguments when dealing with default values or processors.

---

`\l__cmd_args_ii_tl`

```
13 \tl_new:N \l__cmd_args_i_tl
14 \tl_new:N \l__cmd_args_ii_tl
```

---

`\l__cmd_current_arg_int` The number of the current argument being set up: this is used to make sure there are at most 9 arguments, then for creating the expandable auxiliary functions and knowing how many arguments the code function should take.

```
15 \int_new:N \l__cmd_current_arg_int
```

---

\l\_\_cmd\_defaults\_bool The boolean indicates whether there are any argument with default value other than -*NoValue*-; the token list holds the code to determine these default values in terms of other arguments.

```
16 \bool_new:N \l__cmd_defaults_bool
17 \tl_new:N \l__cmd_defaults_tl
```

---

\l\_\_cmd\_environment\_bool Generating environments uses the same mechanism as generating functions. However, full processing of arguments is always needed for environments, and so the function-generating code needs to know this. This variable is also used at run time to give correct error messages.

```
18 \bool_new:N \l__cmd_environment_bool
```

---

\l\_\_cmd\_environment\_str Name of the environment, used at definition time and at run time.

```
19 \str_new:N \l__cmd_environment_str
```

---

\l\_\_cmd\_expandable\_bool Used to indicate if an expandable command is being generated, as this affects both the acceptable argument types and how they are implemented.

```
20 \bool_new:N \l__cmd_expandable_bool
```

---

\l\_\_cmd\_expandable\_aux\_name\_tl

Used to create pretty-printing names for the auxiliaries: although the immediate definition does not vary, the full expansion does and so it does not count as a constant.

```
21 \tl_new:N \l__cmd_expandable_aux_name_tl
22 \tl_set:Nn \l__cmd_expandable_aux_name_tl
23 {
24 \l__cmd_function_tl \c_space_tl
25 (arg~ \int_use:N \l__cmd_current_arg_int)
26 }
```

---

\g\_\_cmd\_grabber\_int Used (in exceptional cases) to get unique names for grabbers used by expandable commands.

```
27 \int_new:N \g__cmd_grabber_int
```

---

\l\_\_cmd\_fn\_tl For passing the pre-formed name of the auxiliary to be used as the parsing function.

```
28 \tl_new:N \l__cmd_fn_tl
```

---

\l\_\_cmd\_fn\_code\_tl For passing the pre-formed name of the auxiliary that contains the actual code.

```
29 \tl_new:N \l__cmd_fn_code_tl
```

\l\_\_cmd\_function\_tl Holds the control sequence name of the function currently being defined: used to avoid passing this as an argument and to avoid repeated use of \cs\_to\_str:N.

<sup>30</sup> \tl\_new:N \l\_\_cmd\_function\_tl

### \l\_\_cmd\_grab\_expandably\_bool

When defining a non-expandable command, indicates whether the arguments can all safely be grabbed by expandable grabbers. This is to support abuses of *xparse* that use protected functions inside csname constructions.

<sup>31</sup> \bool\_new:N \l\_\_cmd\_grab\_expandably\_bool

### \l\_\_cmd\_obey\_spaces\_bool

For trailing optionals.

<sup>32</sup> \bool\_new:N \l\_\_cmd\_obey\_spaces\_bool

\l\_\_cmd\_last\_delimiters\_tl Holds the delimiters (first tokens) of all optional arguments since the previous mandatory argument, to warn about cases where it would be impossible to omit optional arguments completely because the following mandatory argument has the same delimiter as one of the optional arguments.

<sup>33</sup> \tl\_new:N \l\_\_cmd\_last\_delimiters\_tl

### \l\_\_cmd\_long\_bool

Used to indicate that an argument is long, on a per-argument basis.

<sup>34</sup> \bool\_new:N \l\_\_cmd\_long\_bool

### \l\_\_cmd\_suppress\_strip\_bool

Used to indicate that an a pair of braces should not be stripped from an optional argument.

<sup>35</sup> \bool\_new:N \l\_\_cmd\_suppress\_strip\_bool

\l\_\_cmd\_m\_args\_int The number of m arguments: if this is the same as the total number of arguments, then a short-cut can be taken in the creation of the grabber code.

<sup>36</sup> \int\_new:N \l\_\_cmd\_m\_args\_int

\l\_\_cmd\_prefixed\_bool When preparing the signature of non-expandable commands, indicates that the current argument is affected by a processor or by + (namely is long).

<sup>37</sup> \bool\_new:N \l\_\_cmd\_prefixed\_bool

---

```
\l__cmd_process_all_t1
\l__cmd_process_one_t1
\l__cmd_process_some_bool
```

When preparing the signature, the processors that will be applied to a given argument are collected in `\l__cmd_process_one_t1`, while `\l__cmd_process_all_t1` contains processors for all arguments. The boolean indicates whether there are any processors (to bypass the whole endeavour otherwise).

```
38 \tl_new:N \l__cmd_process_all_t1
39 \tl_new:N \l__cmd_process_one_t1
40 \bool_new:N \l__cmd_process_some_bool
```

---

```
\l__cmd_saved_args_t1
```

Stores `\l__cmd_args_t1` to deal with space-trimming of b-type arguments.

```
41 \tl_new:N \l__cmd_saved_args_t1
```

---

```
\l__cmd_signature_t1
```

Used when constructing the signature (code for argument grabbing) to hold what will become the implementation of the main function. When arguments are grabbed (at point of use of the command/environment), it also stores the code for grabbing the remaining arguments.

```
42 \tl_new:N \l__cmd_signature_t1
```

---

```
\l__cmd_some_obey_spaces_bool
\l__cmd_some_long_bool
\l__cmd_some_short_bool
```

These flags are set while normalizing the argument specification. The `obey_spaces` one is used to detect when ! is used on an argument that is not a trailing optional argument. The other two are used to check whether all short arguments appear before long arguments: this is needed to grab arguments expandably. As soon as the first long argument is seen (other than t-type, whose long status is ignored) the `some_long` flag is set. The `some_short` flag is used for expandable commands, to know whether to define a short auxiliary too.

```
43 \bool_new:N \l__cmd_some_obey_spaces_bool
44 \bool_new:N \l__cmd_some_long_bool
45 \bool_new:N \l__cmd_some_short_bool
```

---

```
\q__cmd_recursion_tail
```

Quarks and functions for internal processing.

```
46 \quark_new:N \q__cmd_recursion_tail
47 \quark_new:N \q__cmd_recursion_stop
48 _kernel_quark_new_test:N _cmd_if_recursion_tail_stop_do:Nn
```

(End of definition for `\_cmd_if_recursion_tail_stop_do:Nn` and  
`\_cmd_use_i_delimit_by_q_recursion_stop:nw`.)

---

```
\l__cmd_tmp_prop
\l_\cmdtmpapt\ Scratch space.
\l__cmd_tmpb_tl
```

49 \prop\_new:N \l\_\_cmd\_tmp\_prop  
50 \tl\_new:N \l\_\_cmd\_tmpa\_tl  
51 \tl\_new:N \l\_\_cmd\_tmpb\_tl  
52 \cs\_new\_eq:NN \\_\_cmd\_tmp:w ?

(End of definition for \\_\_cmd\_tmp:w.)

With `xparse`, information about commands being (re)defined was switched off by default, unless the `log-declarations` package option was used, so here we'll switch that off as well.

```
53 \msg_redirect_module:nnn { cmd } { info } { none }
Also add cmd to the LaTeX messages.
54 \prop_gput:Nnn \g_msg_module_type_prop { cmd } { LaTeX }
```

## 1.2 Declaring commands and environments

The main functions for creating commands set the appropriate flag then use the same internal code to do the definition.

```
55 \cs_new_protected:Npn __cmd_declare_cmd:Nnn
56 {
57 \bool_set_false:N \l__cmd_expandable_bool
58 __cmd_declare_cmd_aux:Nnn
59 }
60 \cs_new_protected:Npn __cmd_declare_expandable_cmd:Nnn
61 {
62 \bool_set_true:N \l__cmd_expandable_bool
63 __cmd_declare_cmd_aux:Nnn
64 }
```

The first stage is to log information, both for the user in the log and for programmatic use in a property list of all declared commands.

```
65 \cs_new_protected:Npn __cmd_declare_cmd_aux:Nnn #1#2#3
66 {
67 \cs_if_exist:NTF #1
68 {
69 \msg_info:nnxx { cmd } { redefine }
70 { \token_to_str:N #1 } { \tl_to_str:n {#2} }
71 }
72 {
73 \bool_lazy_or:nnT
74 { \cs_if_exist_p:c { \cs_to_str:N #1 ~ code } }
75 { \cs_if_exist_p:c { \cs_to_str:N #1 ~ defaults } }
76 {
77 \msg_warning:nnx { cmd } { unsupported-let }
78 { \token_to_str:N #1 }
79 }
80 \msg_info:nnxx { cmd } { define-command }
81 { \token_to_str:N #1 } { \tl_to_str:n {#2} }
82 }
83 \bool_set_false:N \l__cmd_environment_bool
84 __cmd_declare_cmd_internal:Nnnn #1 {#2} {#3} { }
```

```
85 }
```

At definition time, the variable `\l__cmd_fn_tl` is only used for error messages. The real business of defining a document command starts with setting up the appropriate name, then normalizing the argument specification to get rid of shorthands.

```
86 \cs_new_protected:Npn __cmd_declare_cmd_internal:Nnnn #1#2#3#4
87 {
88 \tl_set:Nx \l__cmd_function_tl { \cs_to_str:N #1 }
89 __cmd_normalize_arg_spec:n {#2}
90 \exp_args:No __cmd_prepare_signature:n \l__cmd_arg_spec_tl
91 __cmd_declare_cmd_code:Nnn #1 {#2} {#3}
92 #4
93 __cmd_break_point:n {#2}
94 }
```

(End of definition for `\__cmd_declare_cmd:Nnn` and others.)

`\__cmd_break_point:n` A marker used to escape from creating a definition if necessary.

```
95 \cs_new_eq:NN __cmd_break_point:n \use_none:n
```

(End of definition for `\__cmd_break_point:n`.)

`\__cmd_all_m_check:n` A quick loop to check for all (+)m-type arguments.

```
96 \cs_new:Npn __cmd_all_m_check:n #1
97 { \tl_map_function:nN {#1} __cmd_all_m_check_aux:n }
98 \cs_new:Npn __cmd_all_m_check_aux:n #1
99 {
100 \str_if_eq:nnF {#1} { m }
101 {
102 \str_if_eq:nnF {#1} { + }
103 { X }
104 }
105 }
```

(End of definition for `\__cmd_all_m_check:n` and `\__cmd_all_m_check_aux:n`.)

At this stage we can check for a short-cut possibility: if the argument specification is made up of just (+)m tokens, and if all arguments are either short or long, then we can produce an optimized document command. This only applies to document commands, not creation of environments (which are more complex).

```
106 \cs_new_protected:Npn __cmd_declare_cmd_code:Nnn #1#2
107 {
108 \bool_lazy_any:nTF
109 {
110 { \l__cmd_environment_bool }
111 {
112 \bool_lazy_and_p:nn
113 { \l__cmd_some_short_bool }
114 { \l__cmd_some_long_bool }
115 }
116 { ! \tl_if_blank_p:e { __cmd_all_m_check:n {#2} } }
117 }
118 {
119 \tl_set:Nx \l__cmd_fn_tl
120 { \exp_not:c { \l__cmd_function_tl \c_space_tl } }
```

```

121 \bool_if:NTF \l__cmd_grab_expandably_bool
122 { __cmd_declare_cmd_code_expandable:Nnn }
123 { __cmd_declare_cmd_code_aux:Nnn }
124 }
125 { __cmd_declare_cmd_optimized:Nnn }
126 #1 {#2}
127 }

```

The optimized version of commands just has to worry about whether to make them protected or long. The commands start with an expandable marker so that other parts of the kernel know these are set up by \tcmd. We need the two layers of redirection so that the `code` internal function has the same form as it would for any other document command. Optimization means that there is no `\group_align_safe_begin:` before grabbing the arguments, so anything involving & tokens will not work. However, this is really only intended for making optional argument processing safe anyway, so in practice should not be an issue.

```

128 \cs_new_protected:Npn __cmd_declare_cmd_optimized:Nnn #1#2#3
129 {
130 \bool_if:NTF \l__cmd_expandable_bool
131 { \cs_set_nopar:Npe }
132 { \cs_set_protected_nopar:Npe }
133 #1
134 {
135 \exp_not:N __cmd_start_optimized:
136 \exp_not:c { \l__cmd_function_tl \c_space_tl code }
137 }
138 \exp_args:Ncc \cs_generate_from_arg_count:NNnn
139 { \l__cmd_function_tl \c_space_tl code }
140 {
141 cs_set
142 \bool_if:NF \l__cmd_expandable_bool { _protected }
143 \bool_if:NF \l__cmd_some_long_bool { _nopar }
144 :Npn
145 }
146 \l__cmd_current_arg_int
147 {#3}
148 }
149 \cs_new:Npn __cmd_start_optimized: { }

```

Standard functions call `\__cmd_start:nNNnnn`, which receives the argument specification, an auxiliary used for grabbing arguments, an auxiliary containing the code, and then the signature, default arguments, and processors.

```

150 \cs_new_protected:Npn __cmd_declare_cmd_code_aux:Nnn #1#2#3
151 {
152 \cs_generate_from_arg_count:cNnn
153 { \l__cmd_function_tl \c_space_tl code }
154 \cs_set_protected:Npn \l__cmd_current_arg_int {#3}
155 \cs_set_protected_nopar:Npx #1
156 {
157 \bool_if:NTF \l__cmd_environment_bool
158 {
159 __cmd_start_env:nnnnn { \exp_not:n {#2} }
160 { \l__cmd_environment_str }
161 }

```

```

162 {
163 __cmd_start:nNNnnn { \exp_not:n {#2} }
164 \exp_not:c { \l__cmd_function_tl \c_space_tl }
165 \exp_not:c { \l__cmd_function_tl \c_space_tl code }
166 }
167 { \exp_not:o \l__cmd_signature_tl }
168 {
169 \bool_if:NT \l__cmd_defaults_bool
170 { \exp_not:o \l__cmd_defaults_tl }
171 }
172 {
173 \bool_if:NT \l__cmd_process_some_bool
174 { \exp_not:o \l__cmd_process_all_tl }
175 }
176 }
177 }

```

Expandable functions and functions whose arguments can be grabbed expandably call `\__cmd_start_expandable:nNNNNn`, which receives the argument specification, four auxiliaries (two for grabbing arguments, one for the code, and one for default arguments), and finally the signature. Non-expandable functions that take this branch should nevertheless be protected, as well as their `code` function. They will only be expanded in contexts such as constructing a `cname`. The two grabbers (named after the function with one or two spaces) are needed when there are both short and long arguments; otherwise the same grabber is included twice in the definition. If all arguments are long or all are short (only) grabber is defined correspondingly to be long/short. Otherwise two grabbers are defined, one long, one short.

```

178 \cs_new_protected:Npn __cmd_declare_cmd_code_expandable:Nnn #1#2#3
179 {
180 \exp_args:Ncc \cs_generate_from_arg_count:NNnn
181 { \l__cmd_function_tl \c_space_tl code }
182 { cs_set \bool_if:NF \l__cmd_expandable_bool { _protected } :Npn }
183 \l__cmd_current_arg_int {#3}
184 \bool_if:NT \l__cmd_defaults_bool
185 {
186 \use:x
187 {
188 \cs_generate_from_arg_count:cNnn
189 { \l__cmd_function_tl \c_space_tl defaults }
190 \cs_set:Npn \l__cmd_current_arg_int
191 { \exp_not:o \l__cmd_defaults_tl }
192 }
193 }
194 \bool_if:NTF \l__cmd_expandable_bool
195 { \cs_set_nopar:Npx } { \cs_set_protected_nopar:Npx } #1
196 {
197 \exp_not:N __cmd_start_expandable:nNNNNn
198 { \exp_not:n {#2} }
199 \exp_not:c { \l__cmd_function_tl \c_space_tl }
200 \exp_not:c
201 {
202 \l__cmd_function_tl \c_space_tl
203 \bool_if:NT \l__cmd_some_short_bool
204 { \bool_if:NT \l__cmd_some_long_bool { \c_space_tl } }

```

```

205 }
206 \exp_not:c { \l__cmd_function_tl \c_space_tl code }
207 \bool_if:NTF \l__cmd_defaults_bool
208 { \exp_not:c { \l__cmd_function_tl \c_space_tl defaults } }
209 { ? }
210 { \exp_not:o \l__cmd_signature_tl }
211 }
212 \bool_if:NTF \l__cmd_some_long_bool
213 {
214 \bool_if:NT \l__cmd_some_short_bool
215 {
216 \cs_set_nopar:cpx { \l__cmd_function_tl \c_space_tl \c_space_tl }
217 ##1##2 { ##1 {##2} }
218 }
219 \cs_set:cpx
220 }
221 { \cs_set_nopar:cpx }
222 { \l__cmd_function_tl \c_space_tl } ##1##2 { ##1 {##2} }
223 }

```

(End of definition for `\__cmd_declare_cmd_code:Nnn` and others.)

The lead-off to creating an environment is much the same as that for creating a command: issue the appropriate message, store the argument specification then hand off to an internal function.

```

__cmd_declare_env:nnnn
__cmd_declare_env:ennn
 \cmd_declare_env_internal:nnnn
__cmd_set_environment_end:n
__cmd_set_environment_end:n

224 <latexrelease> \IncludeInRelease{2024/11/01}{__cmd_declare_env:nnnn}%
225 <latexrelease> {Use~space-trimmed~envname~directly}
226 \cs_new_protected:Npn __cmd_declare_env:nnnn #1#2
227 {
228 \str_set:Nn \l__cmd_environment_str {#1}
229 \cs_if_exist:cTF { #1 }
230 { \msg_info:nnnn { cmd } { redefine-env } { #1 } { #2 } }
231 { \msg_info:nnnn { cmd } { define-env } { #1 } { #2 } }
232 \bool_set_false:N \l__cmd_expandable_bool
233 \bool_set_true:N \l__cmd_environment_bool
234 __cmd_declare_env_internal:nnnn {#1} {#2}
235 }
236 \cs_generate_variant:Nn __cmd_declare_env:nnnn { e }
237 <latexrelease> \EndIncludeInRelease
238 <latexrelease> \IncludeInRelease{2024/06/01}{__cmd_declare_env:nnnn}%
239 <latexrelease> {Use~space-trimmed~envname~directly}
240 <latexrelease> \cs_new_protected:Npn __cmd_declare_env:nnnn #1#2
241 <latexrelease> {
242 \str_set:Nx \l__cmd_environment_str {#1}
243 \str_set:Nx \l__cmd_environment_str
244 { \tl_trim_spaces:o { \l__cmd_environment_str } }
245 \cs_if_exist:cTF { \l__cmd_environment_str }
246 {
247 \msg_info:nnxx { cmd } { redefine-env }
248 { \l__cmd_environment_str } { \tl_to_str:n {#2} }
249 }
250 {
251 \msg_info:nnxx { cmd } { define-env }
252 { \l__cmd_environment_str } { \tl_to_str:n {#2} }

```

```

253 <|latexrelease> }
254 <|latexrelease> \bool_set_false:N \l__cmd_expandable_bool
255 <|latexrelease> \bool_set_true:N \l__cmd_environment_bool
256 <|latexrelease> \exp_args:NV \l__cmd_declare_env_internal:nnnn
257 <|latexrelease> \l__cmd_environment_str {\#2}
258 <|latexrelease> }
259 <|latexrelease>
260 <|latexrelease>\EndIncludeInRelease

```

Creating a document environment requires a few more steps than creating a single command. In order to pass the arguments of the command to the end of the function, it is necessary to store the grabbed arguments. To do that, the function used at the end of the environment has to be redefined to contain the appropriate information. To minimize the amount of expansion at point of use, the code here is expanded now as well as when used. The last argument of `\l__cmd_declare_cmd_internal:Nnnn` is only run if the definition succeeded. In package mode this ensures that the original definition of the environment is not changed if the definition fails for any reason. This also avoids an error when defining the `end_aux` function when the user asks for more than 9 arguments.

```

261 \cs_new_protected:Npn \l__cmd_declare_env_internal:nnnn #1#2#3#4
262 {
263 \exp_args:Nc \l__cmd_declare_cmd_internal:Nnnn { environment~ #1 } {\#2}
264 {\#3}
265 {
266 \cs_set_nopar:cpx { environment~ #1 ~end }
267 { \exp_not:c { environment~ #1 ~end-aux } }
268 \cs_generate_from_arg_count:cNnn
269 { environment~ #1 ~end-aux~ } \cs_set:Npn
270 \l__cmd_current_arg_int {\#4}
271 \cs_set_eq:cc {\#1} { environment~ #1 }
272 \cs_set_eq:cc { end #1 } { environment~ #1 ~end }
273 }
274 }
275 \cs_new_protected:Npn \l__cmd_set_environment_end:n #1
276 {
277 \cs_set_nopar:cpx { environment~ #1 ~end-aux }
278 {
279 \exp_not:c { environment~ #1 ~end-aux~ }
280 \exp_not:o \l__cmd_args_tl
281 }
282 }

```

(End of definition for `\l__cmd_declare_env:nnnn`, `\l__cmd_declare_env_internal:nnnn`, and `\l__cmd_set_environment_end:n`.)

### 1.3 Structure of xparse commands

```
\l__cmd_start_env:nnnnn
\l__cmd_start:nNNnnn
```

For error messages that occur during run-time when getting arguments of environments it is necessary to keep track of the environment name. We begin non-expandable commands with a token equal to `\scan_stop:`, whose name gives a reasonable error message if the command is used inside a csname and protects against f-expansion. This is useless for environments since `\begin` is already not expandable. Both the command and environment codes start with `\group_align_safe_begin:`, then `\l__cmd_run_code:` (used

by both) does `\group_align_safe_end`, so that delimited arguments may be grabbed in alignments if they contain and alignment tab token (see [latex3/latex3/issues/839](#)).

```

283 \cs_new_protected:Npn __cmd_start_env:nnnnn #1#2
284 {
285 \conditionally@traceoff
286 \group_align_safe_begin:
287 \str_set:Nn \l__cmd_environment_str {#2}
288 \bool_set_true:N \l__cmd_environment_bool
289 __cmd_start_aux:ccnnnn
290 { environment~\l__cmd_environment_str \c_space_tl }
291 { environment~\l__cmd_environment_str \c_space_tl code }
292 {#1}
293 }
294 \cs_new_protected:Npx __cmd_start:nNNnnn #1#2#3
295 {
296 \exp_not:c { xparse~function~is~not~expandable }
297 \exp_not:N \conditionally@traceoff
298 \exp_not:N \group_align_safe_begin:
299 \exp_not:n { \bool_set_false:N \l__cmd_environment_bool }
300 \exp_not:N __cmd_start_aux:NNnnn
301 #2 #3 {#1}
302 }
```

*(End of definition for `\__cmd_start_env:nnnnn` and `\__cmd_start:nNNnnn`.)*

`\__cmd_start_aux:NNnnn`  
`\__cmd_start_aux:ccnnnn`

This sets up a few variables to minimize the boilerplate code included in all `xparse`-defined commands. It then runs the grabbers `#4`. Again, the argument specification `#1` is only for diagnostics.

```

303 \cs_new_protected:Npn __cmd_start_aux:NNnnnn #1#2#3#4#5#6
304 {
305 \tl_clear:N \l__cmd_args_tl
306 \tl_set:Nn \l__cmd_fn_tl {#1}
307 \tl_set:Nn \l__cmd_fn_code_tl {#2}
308 \tl_set:Nn \l__cmd_defaults_tl {#5}
309 \tl_set:Nn \l__cmd_process_all_tl {#6}
310 #4
311 __cmd_run_code:
312 }
313 \cs_generate_variant:Nn __cmd_start_aux:NNnnnn { cc }
```

*(End of definition for `\__cmd_start_aux:NNnnnn`.)*

`\__cmd_run_code:` After arguments are grabbed, this function is responsible for inserting default values, running processors, and finally doing `\group_align_safe_end`: as promised, and running the code.

```

314 \cs_new_protected:Npn __cmd_run_code:
315 {
316 \tl_if_empty:NF \l__cmd_defaults_tl { __cmd_defaults: }
317 \tl_if_empty:NF \l__cmd_process_all_tl { __cmd_args_process: }
318 \bool_if:NT \l__cmd_environment_bool
319 { \exp_args:No __cmd_set_environment_end:n \l__cmd_environment_str }
320 \group_align_safe_end:
321 \conditionally@traceon
322 \exp_after:wN \l__cmd_fn_code_tl \l__cmd_args_tl
323 }
```

(End of definition for `\_cmd_run_code`.)

```
_cmd_defaults:
_cmd_defaults_def:
_cmd_defaults_def:nn
_cmd_defaults_def:nnn
_cmd_defaults_aux:
_cmd_defaults_error:w
```

First construct `\_cmd_tmp:w` (see below) that will receive the arguments found so far and determine default values for any missing argument. Then call it repeatedly until the set of arguments stabilizes. Since that could lead to an infinite loop we only call it up to nine times, the maximal number needed for stabilization if there is a chain of arguments that depend on each other. If that fails to stabilize raise an error.

```
324 \cs_new_protected:Npn _cmd_defaults:
325 {
326 _cmd_defaults_def:
327 \tl_set_eq:NN \l__cmd_args_i_tl \l__cmd_args_tl
328 _cmd_defaults_aux: _cmd_defaults_aux: _cmd_defaults_aux:
329 _cmd_defaults_aux: _cmd_defaults_aux: _cmd_defaults_aux:
330 _cmd_defaults_aux: _cmd_defaults_aux: _cmd_defaults_aux:
331 _cmd_defaults_error:w
332 \q_recursion_stop
333 \tl_set_eq:NN \l__cmd_args_tl \l__cmd_args_i_tl
334 }
335 \cs_new_protected:Npn _cmd_defaults_aux:
336 {
337 \tl_set:Nx \l__cmd_args_ii_tl
338 { \exp_after:wN _cmd_tmp:w \l__cmd_args_i_tl }
339 \tl_if_eq:NNT \l__cmd_args_ii_tl \l__cmd_args_i_tl
340 { \use_none_delimit_by_q_recursion_stop:w }
341 \tl_set_eq:NN \l__cmd_args_i_tl \l__cmd_args_ii_tl
342 }
343 \cs_new_protected:Npn _cmd_defaults_error:w \q_recursion_stop
344 {
345 \msg_error:nnx { cmd } { default-loop }
346 { _cmd_environment_or_command: }
347 }
```

To construct `\_cmd_tmp:w`, first go through the arguments found and the corresponding defaults, building a token list with `{#<arg number>}` for arguments found in the input (whose default will not be used) and otherwise `{\exp_not:n{<default>}}` for arguments whose default will be used.

```
348 \cs_new_protected:Npn _cmd_defaults_def:
349 {
350 \tl_clear:N \l__cmd_tmptl
351 \int_zero:N \l__cmd_current_arg_int
352 _cmd_tmpthread_function:NNN \l__cmd_args_tl \l__cmd_defaults_tl
353 _cmd_defaults_def:nn
354 \cs_generate_from_arg_count:NNno _cmd_tmp:w \cs_set:Npn
355 \l__cmd_current_arg_int \l__cmd_tmptl
356 }
357 \cs_generate_variant:Nn \cs_generate_from_arg_count:NNnn { NNno }
358 \cs_new_protected:Npn _cmd_defaults_def:nn
359 {
360 \int_incr:N \l__cmd_current_arg_int
361 \exp_args:NV _cmd_defaults_def:nnn \l__cmd_current_arg_int
362 }
363 \cs_new_protected:Npn _cmd_defaults_def:nnn #1#2#3
364 {
365 \tl_put_right:Nx \l__cmd_tmptl
```

```

366 {
367 {
368 \exp_not:N \exp_not:n
369 {
370 \tl_if_novalue:nTF {#2}
371 { \exp_not:o {#3} }
372 { \exp_not:n { ## #1 } }
373 }
374 }
375 }
376 }
```

(End of definition for `\__cmd_defaults:` and others.)

```
__cmd_args_process:
__cmd_args_process_loop:nn
__cmd_args_process_aux:n
```

Loop through arguments (stored in `\l__cmd_args_tl`) and the corresponding processors (in `\l__cmd_process_all_tl`) simultaneously, apply all processors for each argument and store the result back into `\l__cmd_args_tl`. To allow processors to depend on other arguments, for every processor define a temporary auxiliary that receives all arguments `\l__cmd_args_tl`.

```

377 \cs_new_protected:Npn __cmd_args_process:
378 {
379 \tl_clear:N \l__cmd_args_ii_tl
380 __cmd_tl_mapthread_function:NNN
381 \l__cmd_args_tl
382 \l__cmd_process_all_tl
383 __cmd_args_process_loop:nn
384 \tl_set_eq:NN \l__cmd_args_tl \l__cmd_args_ii_tl
385 }
386 \cs_new_protected:Npn __cmd_args_process_loop:nn #1#2
387 {
388 \tl_set:Nn \ProcessedArgument {#1}
389 \tl_if_novalue:nF {#1}
390 { \tl_map_function:nN {#2} __cmd_args_process_aux:n }
391 \tl_put_right:No \l__cmd_args_ii_tl
392 { \exp_after:wN \ProcessedArgument } }
393 }
394 \cs_new_protected:Npn __cmd_args_process_aux:n #1
395 {
396 \cs_generate_from_arg_count:NNnn __cmd_tmp:w \cs_set:Npn
397 { \tl_count:N \l__cmd_args_tl } {#1}
398 \exp_args:NNNo \exp_after:wN __cmd_tmp:w \l__cmd_args_tl
399 { \ProcessedArgument }
400 }
```

(End of definition for `\__cmd_args_process:`, `\__cmd_args_process_loop:nn`, and `\__cmd_args_process_aux:n`.)

```
__cmd_start_expandable:nNNNNn
```

This is called for all expandable commands. #6 is the signature, responsible for grabbing arguments. #5 is used to determine default values (or is ? if there are none). #4 is the code to run. #2 and #3 are functions (named after the command) that grab a single argument in the input stream (#3 is short). The argument specification #1 is only used by diagnostic functions. Same as for the non-expandable version, this starts with `\group_align_safe_begin:`, which expands to nothing, so may be safely used in an expandable context.

```

401 \cs_new:Npn __cmd_start_expandable:nNNNNn #1#2#3#4#5#6
402 {
403 \group_align_safe_begin:
404 #6 __cmd_end_expandable:NNw #5 #4 \q__cmd #2#3
405 }

```

(End of definition for `\__cmd_start_expandable:nNNNNn`.)

```

__cmd_end_expandable:NNw
__cmd_end_expandable_aux:w
 __cmd_end_expandable_aux:nNNNn
__cmd_end_expandable_defaults:nnNNn
 __cmd_end_expandable_defaults:nnw
 __cmd_end_expandable_defaults:nw

```

Followed by a function #1 to determine default values (or ? if there are no defaults), the code #2, arguments that have been grabbed, then `\q__cmd` and two generic grabbers. The idea to find default values is similar to the non-expandable case but we cannot define an auxiliary function, so at every step in the loop we need to go through all arguments searching for which ones started out as `-NoValue-` and replacing these by the newly computed values. In fact we need to keep track of three versions of all arguments: the original version, the previous version with default values, and the currently built version (first argument of `\__cmd_end_expandable_defaults:nnnNNn`).

```

406 \cs_new:Npn __cmd_end_expandable:NNw #1#2
407 { __cmd_end_expandable_aux:w #1#2 \prg_do_nothing: }
408 \cs_new:Npn __cmd_end_expandable_aux:w #1#2#3 \q__cmd
409 { \exp_args:No __cmd_end_expandable_aux:nNNNN {#3} #1 #2 }
410 \cs_new:Npn __cmd_end_expandable_aux:nNNNN #1#2#3#4#5
411 {
412 \token_if_eq_charcode:NNT ? #2 { \exp_after:wN \use_iv:nnnn }
413 __cmd_end_expandable_defaults:nnnNNn {#1} { } {#1} #2#3
414 { } { } { } { } { } { } { } { } { }
415 {
416 \msg_expandable_error:nnf { cmd } { default-loop }
417 { \exp_args:Nf \tl_trim_spaces:n { \token_to_str:N #4 } }
418 \use_iv:nnnn
419 }
420 \q_stop
421 }
422 \cs_new:Npn __cmd_end_expandable_defaults:nnnNNn #1#2#3#4#5#6
423 {
424 #6
425 \str_if_eq:nnTF {#1} {#2}
426 { \use_i_delimit_by_q_stop:nw { \group_align_safe_end: #5 #1 } }
427 {
428 \exp_args:No __cmd_tl_mapthread_function:nnN
429 { #4 #1 } {#3}
430 __cmd_end_expandable_defaults:nnw
431 __cmd_end_expandable_defaults:nnnNNn { } {#1} {#3} #4 #5
432 }
433 }
434 \cs_new:Npn __cmd_end_expandable_defaults:nnw #1#2
435 {
436 \tl_if_novalue:nTF {#2}
437 { \exp_args:No __cmd_end_expandable_defaults:nw {#1} }
438 { __cmd_end_expandable_defaults:nw {#2} }
439 }
440 \cs_new:Npn __cmd_end_expandable_defaults:nw
441 #1#2 __cmd_end_expandable_defaults:nnnNNn #3
442 { #2 __cmd_end_expandable_defaults:nnnNNn { #3 {#1} } }

(End of definition for __cmd_end_expandable:NNw and others.)
```

## 1.4 Normalizing the argument specifications

The goal here is to expand aliases and check that the argument specification is valid before the main parsing run. If it is not valid the entire set up is abandoned to avoid any strange internal errors. A function is provided for each argument type that will grab any extra data items and call the loop function after performing the following checks and tasks.

- Check that each argument has the correct number of data items associated with it, and that where a single character is required, one has actually been supplied.
- Check that processors and the markers +, ! and = are followed by an argument for which they make sense, and are not redundant.
- Check the absence of forbidden types for expandable commands, namely G/v always, and 1/u after optional arguments (`xparse` may have inserted braces due to a failed search for an optional argument).
- Check that no optional argument is followed by a mandatory argument with the same delimiter, as otherwise the optional argument could never be omitted.
- Keep track in `\l__cmd_some_long_bool` and `\l__cmd_some_short_bool` of whether the command has some long/short arguments.
- Keep track in `\l__cmd_grab_expandably_bool` of whether all arguments are m/1/u type and short arguments appear before long ones, in which case they can be grabbed expandably just as safely as they could be grabbed nonexpandably. Regardless of that, arguments of expandable commands will be grabbed expandably and arguments of environments will not (because the list of arguments built by non-expandable grabbing is used to pass them to the end-environment code).

Further checks happen at the end of the loop:

- that there are at most 9 arguments;
- that an expandable command does not end with an optional argument (this case is detected by using the fact that `\l__cmd_last_delimiters_tl` is cleared by every mandatory argument and filled by every optional argument).

```
__cmd_normalize_arg_spec:n Loop through the argument specification, calling an auxiliary specific to each argument
__cmd_normalize_arg_spec_loop:n type. If any argument is unknown stop the definition.

443 \cs_new_protected:Npn __cmd_normalize_arg_spec:n #1
444 {
445 \int_zero:N \l__cmd_current_arg_int
446 \tl_clear:N \l__cmd_last_delimiters_tl
447 \tl_clear:N \l__cmd_arg_spec_tl
448 \bool_set_true:N \l__cmd_grab_expandably_bool
449 \bool_set_false:N \l__cmd_obey_spaces_bool
450 \bool_set_false:N \l__cmd_long_bool
451 \bool_set_false:N \l__cmd_suppress_strip_bool
452 \bool_set_false:N \l__cmd_some_obey_spaces_bool
453 \bool_set_false:N \l__cmd_some_long_bool
454 \bool_set_false:N \l__cmd_some_short_bool
455 __cmd_normalize_arg_spec_loop:n #1
456 \q_recursion_tail \q_recursion_tail \q_recursion_stop
```

```

457 \int_compare:nNnT \l__cmd_current_arg_int > 9
458 {
459 \msg_error:nnxx { cmd } { too-many-args }
460 { __cmd_environment_or_command: } { \tl_to_str:n {#1} }
461 __cmd_bad_def:wn
462 }
463 \bool_if:NT \l__cmd_expandable_bool
464 {
465 \tl_if_empty:NF \l__cmd_last_delimiters_tl
466 {
467 \msg_error:nnxx { cmd } { expandable-ending-optional }
468 { \iow_char:N \\ \l__cmd_function_tl } { \tl_to_str:n {#1} }
469 __cmd_bad_def:wn
470 }
471 }
472 \bool_if:NT \l__cmd_expandable_bool
473 {
474 \bool_set_true:N \l__cmd_grab_expandably_bool
475 \bool_if:NT \l__cmd_environment_bool
476 { \bool_set_false:N \l__cmd_grab_expandably_bool }
477 }
478 \cs_new_protected:Npn __cmd_normalize_arg_spec_loop:n #1
479 {
480 \quark_if_recursion_tail_stop:n {#1}
481 \int_incr:N \l__cmd_current_arg_int
482 \cs_if_exist_use:cF { __cmd_normalize_type_ \tl_to_str:n {#1} :w }
483 {
484 \bool_lazy_any:nTF
485 {
486 { \str_if_eq_p:nn {#1} { G } }
487 { \str_if_eq_p:nn {#1} { g } }
488 { \str_if_eq_p:nn {#1} { l } }
489 { \str_if_eq_p:nn {#1} { u } }
490 }
491 {
492 \msg_error:nnxx { cmd } { xparse-arg-type }
493 { __cmd_environment_or_command: } { \tl_to_str:n {#1} }
494 }
495 {
496 \msg_error:nnxx { cmd } { unknown-argument-type }
497 { __cmd_environment_or_command: } { \tl_to_str:n {#1} }
498 }
499 __cmd_bad_def:wn
500 }
501 }

```

(End of definition for `\__cmd_normalize_arg_spec:n` and `\__cmd_normalize_arg_spec_loop:n`.)

`\__cmd_normalize_type_d:w`  
`\__cmd_normalize_type_e:w`  
`\__cmd_normalize_type_o:w`  
`\__cmd_normalize_type_0:w`  
`\__cmd_normalize_type_r:w`  
`\__cmd_normalize_type_s:w`

These argument types are aliases of more general ones, for example with the default argument `-NoValue-`. To easily insert that marker expanded in the definitions we call `\__cmd_tmp:w` with the argument `-NoValue-`. For argument types that need additional data, check that the data is present (not `\q_recursion_tail`) before proceeding.

```

501 \cs_set_protected:Npn __cmd_tmp:w #1
502 {
503 \cs_new_protected:Npn __cmd_normalize_type_d:w ##1##2

```

```

504 {
505 \quark_if_recursion_tail_stop_do:nn {##2} { __cmd_bad_arg_spec:wn }
506 __cmd_normalize_type_D:w {##1} {##2} {#1}
507 }
508 \cs_new_protected:Npn __cmd_normalize_type_e:w ##1
509 {
510 \quark_if_recursion_tail_stop_do:nn {##1} { __cmd_bad_arg_spec:wn }
511 __cmd_normalize_type_E:w {##1} { }
512 }
513 \cs_new_protected:Npn __cmd_normalize_type_o:w
514 { __cmd_normalize_type_D:w [] {#1} }
515 \cs_new_protected:Npn __cmd_normalize_type_0:w
516 { __cmd_normalize_type_D:w [] }
517 \cs_new_protected:Npn __cmd_normalize_type_r:w ##1##2
518 {
519 \quark_if_recursion_tail_stop_do:nn {##2} { __cmd_bad_arg_spec:wn }
520 __cmd_normalize_type_R:w {##1} {##2} {#1}
521 }
522 \cs_new_protected:Npn __cmd_normalize_type_s:w
523 { __cmd_normalize_type_t:w * }
524 }
525 \exp_args:No __cmd_tmp:w { \c_novalue_tl }

(End of definition for __cmd_normalize_type_d:w and others.)

```

\\_\_cmd\_normalize\_type\_>:w Check that these prefixes have arguments, namely that the next token is not \q\_recursion\_tail, and remember to leave it after the looping macro. Processors are forbidden in expandable commands. If all is good, store the prefix in the cleaned up \l\_\_cmd\_arg\_spec\_tl, and decrement the argument number as prefixes do not correspond to arguments.

```

526 \cs_new_protected:cpn { __cmd_normalize_type_>:w } #1#2
527 {
528 \quark_if_recursion_tail_stop_do:nn {#2} { __cmd_bad_arg_spec:wn }
529 \bool_if:NT \l__cmd_expandable_bool
530 {
531 \msg_error:nnnx { cmd } { processor-in-expandable }
532 { \iow_char:N \\ \l__cmd_function_tl } { \tl_to_str:n {#1} }
533 __cmd_bad_def:wn
534 }
535 \tl_put_right:Nx \l__cmd_arg_spec_tl { > { \tl_trim_spaces:n {#1} } }
536 \int_decr:N \l__cmd_current_arg_int
537 \bool_set_false:N \l__cmd_grab_expandably_bool
538 __cmd_normalize_arg_spec_loop:n {#2}
539 }
540 \cs_new_protected:cpn { __cmd_normalize_type_+:w } #1
541 {
542 __cmd_normalize_type_aux:NnNn + {#1}
543 \l__cmd_long_bool
544 { \bool_set_true:N \l__cmd_long_bool }
545 }
546 \cs_new_protected:cpn { __cmd_normalize_type_!:w } #1
547 {
548 __cmd_normalize_type_aux:NnNn ! {#1}
549 \l__cmd_obey_spaces_bool

```

```

550 {
551 \bool_set_true:N \l__cmd_obey_spaces_bool
552 \bool_set_true:N \l__cmd_some_obey_spaces_bool
553 }
554 }
555 \cs_new_protected:cpn { __cmd_normalize_type_=:w } #1#2
556 {
557 __cmd_normalize_type_aux:NnNn = {#2}
558 \l__cmd_suppress_strip_bool
559 {
560 \bool_if:NT \l__cmd_expandable_bool
561 {
562 \msg_error:nnxx { cmd } { keyval-in-expandable }
563 { \iow_char:N \\ \l__cmd_function_tl } { \tl_to_str:n {#1} }
564 __cmd_bad_def:wn
565 }
566 \bool_set_true:N \l__cmd_suppress_strip_bool
567 \bool_set_false:N \l__cmd_grab_expandably_bool
568 \tl_put_right:Nx \l__cmd_arg_spec_tl
569 { = { \tl_trim_spaces:n {#1} } }
570 }
571 }
572 \cs_new_protected:Npn __cmd_normalize_type_aux:NnNn #1#2#3#4
573 {
574 \quark_if_recursion_tail_stop_do:nn {#2} { __cmd_bad_arg_spec:wn }
575 \bool_if:NT #3
576 {
577 \msg_error:nnxx { cmd } { two-markers }
578 { __cmd_environment_or_command: } { #1 }
579 __cmd_bad_def:wn
580 }
581 #4
582 \int_decr:N \l__cmd_current_arg_int
583 __cmd_normalize_arg_spec_loop:n {#2}
584 }

```

(End of definition for `\__cmd_normalize_type_>:w` and others.)

`\__cmd_normalize_type_D:w`  
`\__cmd_normalize_type_E:w`  
`\__cmd_normalize_type_t:w`  
`\__cmd_normalize_E_unique_check:w`

Optional argument types. Check that all required data is present (and consists of single characters if applicable) and check for forbidden types for expandable commands. For E-type require that there is at least one embellishment, that each one is a single character, and that there aren't more optional arguments than embellishments; also remember that each embellishment counts as one argument for `\l__cmd_current_arg_int`. Then in each case store the data in `\l__cmd_arg_spec_tl`, and for later checks store in `\l__cmd_last_delimiters_tl` the tokens whose presence determines whether there is an optional argument (for braces store {}, seen later as an empty delimiter).

```

585 \cs_new_protected:Npn __cmd_normalize_type_D:w #1#2#3
586 {
587 \quark_if_recursion_tail_stop_do:nn {#3} { __cmd_bad_arg_spec:wn }
588 __cmd_single_token_check:n {#1} __cmd_allowed_token_check:N #1
589 __cmd_single_token_check:n {#2}
590 __cmd_add_arg_spec:n { D #1 #2 {#3} }
591 \tl_put_right:Nn \l__cmd_last_delimiters_tl {#1}
592 \bool_set_false:N \l__cmd_grab_expandably_bool

```

```

593 __cmd_normalize_arg_spec_loop:n
594 }
595 \cs_new_protected:Npn __cmd_normalize_type_E:w #1#2
596 {
597 \quark_if_recursion_tail_stop_do:nn {#2} { __cmd_bad_arg_spec:wn }
598 \tl_if_blank:nT {#1} { __cmd_bad_arg_spec:wn }
599 \tl_map_function:nN {#1} __cmd_single_token_check:n
600 \tl_map_function:nN {#1} __cmd_allowed_token_check:N
601 __cmd_normalize_E_unique_check:w #1 \q_nil \q_stop
602 \int_compare:nNnT { \tl_count:n {#2} } > { \tl_count:n {#1} }
603 { __cmd_bad_arg_spec:wn }
604 __cmd_add_arg_spec:n { E {#1} {#2} }
605 \tl_put_right:Nn \l__cmd_last_delimiters_tl {#1}
606 \bool_set_false:N \l__cmd_grab_expandably_bool
607 \int_add:Nn \l__cmd_current_arg_int { \tl_count:n {#1} - 1 }
608 __cmd_normalize_arg_spec_loop:n
609 }
610 \cs_new_protected:Npn __cmd_normalize_E_unique_check:w #1#2 \q_stop
611 {
612 \quark_if_nil:NF #1
613 {
614 \tl_if_in:nnT {#2} {#1} { __cmd_bad_arg_spec:wn }
615 __cmd_normalize_E_unique_check:w #2 \q_stop
616 }
617 }
618 \cs_new_protected:Npn __cmd_normalize_type_t:w #1
619 {
620 \quark_if_recursion_tail_stop_do:Nn #1 { __cmd_bad_arg_spec:wn }
621 __cmd_single_token_check:n {#1} __cmd_allowed_token_check:N #1
622 \tl_put_right:Nx \l__cmd_arg_spec_tl
623 {
624 \bool_if:NT \l__cmd_obey_spaces_bool { ! }
625 t \exp_not:n {#1}
626 }
627 \tl_put_right:Nn \l__cmd_last_delimiters_tl {#1}
628 \bool_set_false:N \l__cmd_grab_expandably_bool
629 \bool_set_false:N \l__cmd_obey_spaces_bool
630 \bool_set_false:N \l__cmd_long_bool
631 __cmd_normalize_arg_spec_loop:n
632 }

```

(End of definition for `\__cmd_normalize_type_D:w` and others.)

`\__cmd_normalize_type_m:w`  
`\__cmd_normalize_type_R:w`  
`\__cmd_normalize_type_v:w`

Mandatory arguments. First check the required data is present, consists of single characters where applicable, and that the argument type is allowed for expandable commands if applicable. For the `m` and `R` argument types check that they do not follow some optional argument with that delimiter as otherwise the optional argument could not be omitted. Then save data in `\l__cmd_arg_spec_tl`, count the mandatory argument, and empty the list of last delimiters.

```

633 \cs_new_protected:Npn __cmd_normalize_type_m:w
634 {
635 __cmd_delimiter_check:nnn { } { m } { \iow_char:N \{ }
636 __cmd_add_arg_spec_mandatory:n { m }
637 __cmd_normalize_arg_spec_loop:n

```

```

638 }
639 \cs_new_protected:Npn __cmd_normalize_type_R:w #1#2#3
640 {
641 \quark_if_recursion_tail_stop_do:nn {#3} { __cmd_bad_arg_spec:wn }
642 __cmd_single_token_check:n {#1} __cmd_allowed_token_check:N #1
643 __cmd_single_token_check:n {#2}
644 __cmd_delimiter_check:n {#1} { R/r } { \tl_to_str:n {#1} }
645 \bool_set_false:N \l__cmd_grab_expandably_bool
646 __cmd_add_arg_spec_mandatory:n { R #1 #2 {#3} }
647 __cmd_normalize_arg_spec_loop:n
648 }
649 \cs_new_protected:Npn __cmd_normalize_type_v:w
650 {
651 __cmd_normalize_check_gv:N v
652 __cmd_add_arg_spec_mandatory:n { v }
653 __cmd_normalize_arg_spec_loop:n
654 }

```

*(End of definition for \\_\_cmd\_normalize\_type\_m:w, \\_\_cmd\_normalize\_type\_R:w, and \\_\_cmd\_normalize\_type\_v:w.)*

\\_\_cmd\_normalize\_type\_b:w This argument type is not allowed for commands. This is only allowed at the end of the argument specification, hence we check that #1 is the end.

```

655 \cs_new_protected:Npn __cmd_normalize_type_b:w #1
656 {
657 \bool_if:NF \l__cmd_environment_bool
658 {
659 \msg_error:nnxx { cmd } { invalid-command-arg }
660 { __cmd_environment_or_command: } { b }
661 __cmd_bad_def:wn
662 }
663 \tl_clear:N \l__cmd_last_delimiters_tl
664 __cmd_add_arg_spec:n { b }
665 \quark_if_recursion_tail_stop:n {#1}
666 \msg_error:nnxx { cmd } { arg-after-body }
667 { __cmd_environment_or_command: }
668 { \tl_to_str:n {#1} }
669 __cmd_bad_def:wn
670 }

```

*(End of definition for \\_\_cmd\_normalize\_type\_b:w.)*

\\_\_cmd\_single\_token\_check:n Checks that the argument is a single (non-space) token (possibly surrounded by spaces), and aborts the definition otherwise.

```

671 \cs_new_protected:Npn __cmd_single_token_check:n #1
672 {
673 \tl_trim_spaces_apply:nN {#1} \tl_if_single_token:nF
674 {
675 \msg_error:nnxx { cmd } { not-single-token }
676 { __cmd_environment_or_command: } { \tl_to_str:n {#1} }
677 __cmd_bad_def:wn
678 }
679 }

```

*(End of definition for \\_\_cmd\_single\_token\_check:n.)*

\\_\\_cmd\\_allowed\\_token\\_check:N Some tokens are not allowed as delimiters for some argument types, notably implicit begin/end-group tokens (\bgroup/\egroup). The major problem with these tokens is that for \peek\\_... functions, a literal  $\{_1$  is virtually indistinguishable from a \bgroup or other token which was \let to a  $\{_1$ , and the same goes for  $\}_2$ . All other tokens can be easily distinguished from their implicit counterparts by grabbing them and looking at the string length (see \\_\\_cmd\\_token\\_if\\_cs:NTF), but for begin/end group tokens that is not possible without the risk of mistakenly grabbing the entire brace group (potentially leading to a ! Runaway argument error) or trying to grab a  $\}_2$ , leading to an ! Argument of \dots has an extra } error.

```

680 \cs_new_protected:Npn __cmd_allowed_token_check:N #1
681 {
682 \token_if_eq_meaning:NNTF #1 \c_group_begin_token
683 { \use:n }
684 {
685 \token_if_eq_meaning:NNTF #1 \c_group_end_token
686 { \use:n }
687 { \use_none:n }
688 }
689 {
690 \msg_error:nnxx { cmd } { forbidden-group-token }
691 { __cmd_environment_or_command: } { \tl_to_str:n {#1} }
692 {
693 \token_if_eq_meaning:NNTF #1 \c_group_begin_token
694 { begin } { end }
695 }
696 __cmd_bad_def:wn
697 }
698 }
```

(End of definition for \\_\\_cmd\\_allowed\\_token\\_check:N.)

\\_\\_cmd\_normalize\_check\_gv:N Called for arguments that are always forbidden, or forbidden after an optional argument, for expandable commands.

```

699 \cs_new_protected:Npn __cmd_normalize_check_gv:N #1
700 {
701 \bool_if:NT \l__cmd_expandable_bool
702 {
703 \msg_error:nnxx { cmd } { invalid-expandable-arg }
704 { \iow_char:N \\ \l__cmd_function_tl } { \tl_to_str:n {#1} }
705 __cmd_bad_def:wn
706 }
707 \bool_set_false:N \l__cmd_grab_expandably_bool
708 }
709 \cs_new_protected:Npn __cmd_normalize_check_lu:N #1
710 {
711 \bool_if:NT \l__cmd_expandable_bool
712 {
713 \tl_if_empty:NF \l__cmd_last_delimiters_tl
714 {
715 \msg_error:nnxx { cmd } { invalid-after-optional-expandably }
716 { \iow_char:N \\ \l__cmd_function_tl } { \tl_to_str:n {#1} }
717 __cmd_bad_def:wn
718 }
```

```

719 }
720 }

```

(End of definition for `\_\_cmd\_normalize\_check\_gv:N` and `\_\_cmd\_normalize\_check\_lu:N`.)

`\_\_cmd\_delimiter\_check:nnn` Called for `m` and `R` arguments. Checks that the leading token does not coincide with the token denoting the presence of a previous optional argument. Instead of dealing with braces for the `m`-type we use an empty delimiter to denote that case.

```

721 \cs_new_protected:Npn __cmdDelimiter_check:nnn #1#2#3
722 {
723 \tl_map_inline:Nn \l___cmdLastDelimiters_tl
724 {
725 \tl_if_eq:nnT {##1} {#1}
726 {
727 \msg_warning:nnxx { cmd } { optional-mandatory }
728 {#2} {#3}
729 }
730 }
731 }

```

(End of definition for `\_\_cmd\_delimiter\_check:nnn`.)

`\_\_cmd\_bad\_arg\_spec:wn` `\_\_cmd\_bad\_def:wn` If the argument specification is wrong, this provides an escape from the entire definition process.

```

732 \cs_new_protected:Npn __cmdBadArgSpec:wn #1 __cmdBreakPoint:n #2
733 {
734 \msg_error:nnxx { cmd } { bad-arg-spec }
735 { __cmdEnvironmentOrCommand: } { \tl_to_str:n {#2} }
736 }
737 \cs_new_protected:Npn __cmdBadDef:wn #1 __cmdBreakPoint:n #2 { }

```

(End of definition for `\_\_cmd\_bad\_arg\_spec:wn` and `\_\_cmd\_bad\_def:wn`.)

`\_\_cmd\_add\_arg\_spec:n` `\_\_cmd\_add\_arg\_spec\_mandatory:n` When adding an argument to the argument specification, set the `some\_long` or `some\_short` booleans as appropriate and clear the booleans keeping track of `+`, `!` and `=` markers. Before that, test for a short argument following some long arguments: this is forbidden for expandable commands and prevents grabbing arguments expandably.

For mandatory arguments do some more work, in particular complain if they were preceded by `!`.

```

738 \cs_new_protected:Npn __cmdAddArgSpec:n #1
739 {
740 \bool_lazy_and:nnT
741 { ! \l___cmdLongBool }
742 { \l___cmdSomeLongBool }
743 {
744 \bool_if:NT \l___cmdExpandableBool
745 {
746 \msg_error:nnx { cmd } { long-short-mix }
747 { \iow_char:N \\ \l___cmdFunctionTl }
748 __cmdBadDef:wn
749 }
750 \bool_set_false:N \l___cmdGrabExpandablyBool
751 }
752 \bool_if:NTF \l___cmdLongBool

```

```

753 { \bool_set_true:N \l__cmd_some_long_bool }
754 { \bool_set_true:N \l__cmd_some_short_bool }
755 \tl_put_right:Nx \l__cmd_arg_spec_tl
756 {
757 \bool_if:NT \l__cmd_long_bool { + }
758 \bool_if:NT \l__cmd_obey_spaces_bool { ! }
759 \exp_not:n {\#1}
760 }
761 \bool_set_false:N \l__cmd_long_bool
762 \bool_set_false:N \l__cmd_obey_spaces_bool
763 }
764 \cs_new_protected:Npn __cmd_add_arg_spec_mandatory:n #1
765 {
766 \bool_if:NT \l__cmd_some_obey_spaces_bool
767 {
768 \msg_error:nnnx { cmd } { invalid-bang }
769 { __cmd_environment_or_command: }
770 {
771 \bool_if:NTF \l__cmd_obey_spaces_bool
772 { \tl_to_str:n {\#1} }
773 { an-optional-argument-before~mandatory~ \tl_to_str:n {\#1} }
774 }
775 __cmd_bad_def:wn
776 }
777 \tl_clear:N \l__cmd_last_delimiters_tl
778 __cmd_add_arg_spec:n {\#1}
779 }

```

(End of definition for `\__cmd_add_arg_spec:n` and `\__cmd_add_arg_spec_mandatory:n`.)

## 1.5 Preparing the signature: general mechanism

`\__cmd_prepare_signature:n`  
`\__cmd_prepare_signature:N`  
`\__cmd_prepare_signature_bypass:N`

Actually creating the signature uses the same loop approach as normalizing the signature. There are first a number of variables which need to be set to track what is going on. Many of these variables are unused when defining expandable commands.

```

780 \cs_new_protected:Npn __cmd_prepare_signature:n #1
781 {
782 \int_zero:N \l__cmd_current_arg_int
783 \bool_set_false:N \l__cmd_long_bool
784 \bool_set_false:N \l__cmd_obey_spaces_bool
785 \bool_set_false:N \l__cmd_suppress_strip_bool
786 \int_zero:N \l__cmd_m_args_int
787 \bool_set_false:N \l__cmd_defaults_bool
788 \tl_clear:N \l__cmd_defaults_tl
789 \tl_clear:N \l__cmd_process_all_tl
790 \tl_clear:N \l__cmd_process_one_tl
791 \bool_set_false:N \l__cmd_process_some_bool
792 \tl_clear:N \l__cmd_signature_tl
793 __cmd_prepare_signature:N #1 \q_recursion_tail \q_recursion_stop
794 \bool_if:NF \l__cmd_expandable_bool { __cmd_flush_m_args: }
795 }

```

The main looping function does not take an argument, but carries out the reset on the processor boolean. This is split off from the rest of the process so that when actually setting up processors the flag-reset can be bypassed.

For each known argument type there is an appropriate function to actually do the addition to the signature. These are separate for expandable and standard functions, as the approaches are different.

```

796 \cs_new_protected:Npn __cmd_prepare_signature:N
797 {
798 \bool_set_false:N \l__cmd_prefixed_bool
799 __cmd_prepare_signature_bypass:N
800 }
801 \cs_new_protected:Npn __cmd_prepare_signature_bypass:N #1
802 {
803 \quark_if_recursion_tail_stop:N #1
804 \use:c
805 {
806 __cmd_add
807 \bool_if:NT \l__cmd_grab_expandably_bool { _expandable }
808 _type_ \token_to_str:N #1 :w
809 }
810 }
```

*(End of definition for \\_\_cmd\_prepare\_signature:n, \\_\_cmd\_prepare\_signature:N, and \\_\_cmd\_prepare\_signature\_bypass:N.)*

## 1.6 Setting up a standard signature

Each argument-adding function appends to the signature a grabber (and for some types, the delimiters or default value), except the one for `m` arguments. These are collected and added to the signature all at once by `\__cmd_flush_m_args:`, called for every other argument type. All of the functions then call the loop function `\__cmd_prepare_signature:N`. Default values of arguments are collected by `\__cmd_add_default:n` rather than being stored with the argument; this function and `\__cmd_add_default:` are also responsible for keeping track of `\l__cmd_current_arg_int`.

`\__cmd_add_type_+:w` Making the next argument long means setting the flag. The `m` arguments are recorded here as this has to be done for every case where there is then a long argument.

```

811 \cs_new_protected:cpn { __cmd_add_type_+:w }
812 {
813 __cmd_flush_m_args:
814 \bool_set_true:N \l__cmd_long_bool
815 \bool_set_true:N \l__cmd_prefixed_bool
816 __cmd_prepare_signature_bypass:N
817 }
```

*(End of definition for \\_\_cmd\_add\_type\_+:w.)*

`\__cmd_add_type_!/:w` Much the same for controlling trailing optional arguments.

```

818 \cs_new_protected:cpn { __cmd_add_type_!/:w }
819 {
820 __cmd_flush_m_args:
821 \bool_set_true:N \l__cmd_obey_spaces_bool
822 \bool_set_true:N \l__cmd_prefixed_bool
823 __cmd_prepare_signature_bypass:N
824 }
```

*(End of definition for \\_\_cmd\_add\_type\_!/:w.)*

\\_\\_cmd\\_add\\_type\\_>:w When a processor is found, the processor code is stored. It will be used by \\_\\_cmd\\_args\\_process: once arguments are all found. Here too the loop calls \\_\\_cmd\\_prepare\\_signature\\_bypass:N rather than \\_\\_cmd\\_prepare\\_signature:N so that the flag is not reset.

```

825 \cs_new_protected:cpn { __cmd_add_type_>:w } #1
826 {
827 __cmd_flush_m_args:
828 \bool_set_true:N \l__cmd_prefixed_bool
829 \bool_set_true:N \l__cmd_process_some_bool
830 \tl_put_left:Nn \l__cmd_process_one_tl { [#1] }
831 __cmd_prepare_signature_bypass:N
832 }

```

(End of definition for \\_\\_cmd\\_add\\_type\\_>:w.)

\\_\\_cmd\\_add\\_type\\_=: A mix of the ideas from above: set a flag and add a processor.

```

833 \cs_new_protected:cpn { __cmd_add_type_=:w } #1
834 {
835 __cmd_flush_m_args:
836 \bool_set_true:N \l__cmd_prefixed_bool
837 \bool_set_true:N \l__cmd_suppress_strip_bool
838 \bool_set_true:N \l__cmd_process_some_bool
839 \tl_put_left:Nn \l__cmd_process_one_tl
840 { { __cmd_arg_to_keyvalue:nn [#1] } }
841 __cmd_prepare_signature_bypass:N
842 }

```

(End of definition for \\_\\_cmd\\_add\\_type\\_=:.)

\\_\\_cmd\\_add\\_type\\_b:w

```

843 \cs_new_protected:Npn __cmd_add_type_b:w
844 {
845 __cmd_flush_m_args:
846 __cmd_add_default:
847 __cmd_add_grabber:N b
848 __cmd_prepare_signature:N
849 }

```

(End of definition for \\_\\_cmd\\_add\\_type\\_b:w.)

\\_\\_cmd\\_add\\_type\\_D:w

```

850 \cs_new_protected:Npn __cmd_add_type_D:w #1#2#3
851 {
852 __cmd_flush_m_args:
853 __cmd_add_default:n {#3}
854 __cmd_add_grabber:N D
855 \tl_put_right:Nn \l__cmd_signature_tl { #1 #2 }
856 __cmd_prepare_signature:N
857 }

```

(End of definition for \\_\\_cmd\\_add\\_type\\_D:w.)

\\_\\_cmd\\_add\\_type\\_E:w The E-type argument needs a special handling of default values. Since each embellishment is a separate argument, it also needs to replicate the argument processors for each embellishment argument so that the numbers of arguments and processors remain in sync.

```

858 \cs_new_protected:Npn __cmd_add_type_E:w #1#2
859 {
860 __cmd_flush_m_args:
861 __cmd_add_default_E:nn {#1} {#2}
862 \use:x
863 {
864 __cmd_replicate_processor:nn { \tl_count:n {#1} }
865 { \exp_not:o \l__cmd_process_one_tl }
866 }
867 __cmd_add_grabber:N E
868 \tl_put_right:Nn \l__cmd_signature_tl { {#1} }
869 __cmd_prepare_signature:N
870 }

```

*(End of definition for \\_\\_cmd\\_add\\_type\\_E:w.)*

\\_\\_cmd\_replicate\_processor:nn In the command's argument processor signature (the final argument of \\_\\_cmd\\_start:nNNnnn) there is one braced item for each formal argument (up to nine), and in each of these items there is one braced item for each processor (as many as there were processors declared for a given argument). Something like this:

```

{ % argument processors
 { % argument 1
 { processor 1 } { processor 2 } ... { processor n }
 } % end argument 1
 { ... } % argument 2
 :
 { ... } % argument n
} % end argument processors

```

The function \\_\\_cmd\\_add\\_grabber:N adds one single grabber for an argument, and adds the braced item for that one argument. However, in an E-type argument each embellishment requires its own formal argument, so we need to break out of one layer of braces in \l\_\\_cmd\_process\_one\_tl, add copies of the processor as necessary, and then return the removed brace. The function below does just that: it defines \l\_\\_cmd\_process\_one\_tl starting with a }<sub>2</sub> and ending with a {<sub>1</sub>, so that it adds as many processors as needed when x-expanded.

```

871 \cs_new_protected:Npn __cmd_replicate_processor:nn #1 #2
872 {
873 \int_compare:nNnF {#1} > { 1 } { \use_none:nnn }
874 \tl_set:Nx \l__cmd_process_one_tl
875 {
876 \exp_not:n { \exp_not:n {#2} \if_false: { \fi: } }
877 \prg_replicate:nn { #1 - 2 }
878 { \exp_not:n { \exp_not:n { {#2} } } }
879 \exp_not:n { { \if_false: } \fi: \exp_not:n {#2} }

```

```

880 }
881 }

```

(End of definition for `\_\_cmd\_replicate\_processor:nn`.)

`\_\_cmd\_add\_type_m:w` The `m` type is special as short arguments which are not post-processed are simply counted at this stage. Thus there is a check to see if either of these cases apply. If so, a one-argument grabber is added to the signature. On the other hand, if a standard short argument is required it is simply counted at this stage, to be added later using `\_\_cmd\_flush_m_args`:

```

882 \cs_new_protected:Npn __cmd_add_type_m:w
883 {
884 __cmd_add_default:
885 \bool_if:NTF \l__cmd_prefixed_bool
886 { __cmd_add_grabber:N m }
887 { \int_incr:N \l__cmd_m_args_int }
888 __cmd_prepare_signature:N
889 }

```

(End of definition for `\_\_cmd\_add\_type_m:w`.)

`\_\_cmd\_add\_type_R:w` The `R`-type argument is very similar to the `D`-type.

```

890 \cs_new_protected:Npn __cmd_add_type_R:w #1#2#3
891 {
892 __cmd_flush_m_args:
893 __cmd_add_default:n {#3}
894 __cmd_add_grabber:N R
895 \tl_put_right:Nn \l__cmd_signature_tl { #1 #2 }
896 __cmd_prepare_signature:N
897 }

```

(End of definition for `\_\_cmd\_add\_type_R:w`.)

`\_\_cmd\_add\_type_t:w` Setting up a `t` argument means collecting one token for the test, and adding it along with the grabber to the signature.

```

898 \cs_new_protected:Npn __cmd_add_type_t:w #1
899 {
900 __cmd_flush_m_args:
901 __cmd_add_default:
902 __cmd_add_grabber:N t
903 \tl_put_right:Nn \l__cmd_signature_tl {#1}
904 __cmd_prepare_signature:N
905 }

```

(End of definition for `\_\_cmd\_add\_type_t:w`.)

`\_\_cmd\_add\_type_v:w` At this stage, the `v` argument is identical to `l` except that since the grabber may fail to read a verbatim argument we need a default value.

```

906 \cs_new_protected:Npn __cmd_add_type_v:w
907 {
908 __cmd_flush_m_args:
909 \exp_args:No __cmd_add_default:n \c_novalue_tl
910 __cmd_add_grabber:N v
911 __cmd_prepare_signature:N
912 }

```

*(End of definition for \\_\\_cmd\\_add\\_type\\_v:w.)*

\\_\\_cmd\\_flush\\_m\\_args: As m arguments are simply counted, there is a need to add them to the token register in a block. As this function can only be called if something other than m turns up, the flag can be switched here.

```
913 \cs_new_protected:Npn __cmd_flush_m_args:
914 {
915 \int_compare:nNnT \l__cmd_m_args_int > 0
916 {
917 \tl_put_right:Nx \l__cmd_signature_tl
918 { \exp_not:c { __cmd_grab_m_ \int_use:N \l__cmd_m_args_int :w } }
919 \tl_put_right:Nx \l__cmd_process_all_tl
920 { \prg_replicate:nn { \l__cmd_m_args_int } { { } } }
921 }
922 \int_zero:N \l__cmd_m_args_int
923 }
```

*(End of definition for \\_\\_cmd\_flush\_m\_args:.)*

\\_\\_cmd\\_add\\_grabber:N To keep the various checks needed in one place, adding the grabber to the signature is done here. The only questions are whether the grabber should be long or not, and whether to obey spaces. The \l\_\\_cmd\\_obey\\_spaces\\_bool boolean can only be true for trailing optional arguments. In that case spaces will not be ignored when looking for that optional argument.

```
924 \cs_new_protected:Npn __cmd_add_grabber:N #1
925 {
926 \tl_put_right:Nx \l__cmd_signature_tl
927 {
928 \exp_not:c
929 {
930 __cmd_grab_ #1
931 \bool_if:NT \l__cmd_long_bool { _long }
932 \bool_if:NT \l__cmd_obey_spaces_bool { _obey_spaces }
933 \bool_lazy_and:nn
934 { \l__cmd_suppress_strip_bool }
935 { \str_if_eq_p:nn {#1} { D } }
936 { _no_strip }
937 :w
938 }
939 }
940 \bool_set_false:N \l__cmd_long_bool
941 \bool_set_false:N \l__cmd_obey_spaces_bool
942 \bool_set_false:N \l__cmd_suppress_strip_bool
943 \tl_put_right:Nx \l__cmd_process_all_tl
944 {
945 {
946 \if_charcode:w E #1 \use_i:nn \fi:
947 \exp_not:o \l__cmd_process_one_tl
948 }
949 }
950 \tl_clear:N \l__cmd_process_one_tl
951 }
```

*(End of definition for \\_\\_cmd\\_add\\_grabber:N.)*

|                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>\_\_cmd\_add\_default:n \_\_cmd\_add\_default: \_\_cmd\_add\_default_E:nn</pre> | <p>Store the default value of an argument, or rather code that gives that default value (it may involve other arguments). This is <code>\c_novalue_tl</code> for arguments with no actual default or with default <code>-NoValue-</code>; and (in a brace group) <code>\prg_do_nothing:</code> followed by a default value for others. For E-type arguments, pad the defaults #2 with some <code>\c_novalue_tl</code> until there are as many as embellishments #1. These functions are also used when defining expandable commands.</p> <pre>952 \cs_new_protected:Npn \_\_cmd\_add\_default:n #1 953 { 954     \tl_if_novalue:nTF {#1} 955     { \_\_cmd\_add\_default: } 956     { 957         \int_incr:N \l__cmd_current_arg_int 958         \bool_set_true:N \l__cmd_defaults_bool 959         \tl_put_right:Nn \l__cmd_defaults_tl { { \prg_do_nothing: #1 } } 960     } 961 } 962 \cs_new_protected:Npn \_\_cmd\_add\_default: 963 { 964     \int_incr:N \l__cmd_current_arg_int 965     \tl_put_right:Nn \l__cmd_defaults_tl { \c_novalue_tl } 966 } 967 \cs_new_protected:Npn \_\_cmd\_add\_default_E:nn #1#2 968 { 969     \tl_map_function:nN {#2} \_\_cmd\_add\_default:n 970     \prg_replicate:nn 971     { \tl_count:n {#1} - \tl_count:n {#2} } 972     { \_\_cmd\_add\_default: } 973 }</pre> |
|--------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

(End of definition for `\_\_cmd\_add\_default:n`, `\_\_cmd\_add\_default:`, and `\_\_cmd\_add\_default_E:nn`.)

## 1.7 Setting up expandable types

The approach here is not dissimilar to that for standard types, but fewer types are supported. There is also a need to define the per-function auxiliaries: this is done here, while the general grabbers are dealt with later.

|                                               |                                                                                                                                                                                                                                                             |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>\_\_cmd\_add\_expandable\_type_+:w</pre> | <p>We have already checked that short arguments are before long arguments, so <code>\l__cmd\_long_bool</code> only changes from <code>false</code> to <code>true</code> once (and there is no need to reset it after each argument). Continue the loop.</p> |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```
974 \cs_new_protected:cpn { __cmd_add_expandable_type_+:w }
975 {
976 \bool_set_true:N \l__cmd_long_bool
977 __cmd_prepare_signature:N
978 }
```

(End of definition for `\_\_cmd\_add\_expandable\_type_+:w`.)

|                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>\_\_cmd\_add\_expandable\_type_D: \_\_cmd\_add\_expandable\_type_D_aux:NNN \_\_cmd\_add\_expandable\_type_D_aux:NN \_\_cmd\_add\_expandable\_type_D_aux:NN</pre> | <p>The set up for D-type arguments involves constructing a rather complex auxiliary which is used repeatedly when grabbing. There is an auxiliary here so that the R-type can share code readily: #1 is D or R. The <code>_aux:NN</code> auxiliary is needed if the two delimiting tokens are identical: in contrast to the non-expandable route, the grabber here has to act differently for this case.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

979 \cs_new_protected:Npn __cmd_add_expandable_type_D:w
980 { __cmd_add_expandable_type_D_aux:NNNn D }
981 \cs_new_protected:Npn __cmd_add_expandable_type_D_aux:NNNn #1#2#3#4
982 {
983 __cmd_add_default:n {#4}
984 \tl_if_eq:nnTF {#2} {#3}
985 { __cmd_add_expandable_type_D_aux:NN #1 #2 }
986 { __cmd_add_expandable_type_D_aux:NNN #1 #2 #3 }
987 __cmd_prepare_signature:N
988 }
989 \cs_new_protected:Npn __cmd_add_expandable_type_D_aux:NNN #1#2#3
990 {
991 \bool_if:NTF \l__cmd_long_bool
992 { \cs_set:cpx }
993 { \cs_set_nopar:cpx }
994 { \l__cmd_expandable_aux_name_tl } ##1 ##2 #2 ##3 \q__cmd ##4 #3
995 { ##1 {##2} {##3} {##4} }
996 __cmd_add_expandable_grabber:nn {#1}
997 {
998 \exp_not:c { \l__cmd_expandable_aux_name_tl }
999 \exp_not:n { #2 #3 }
1000 }
1001 }
1002 \cs_new_protected:Npn __cmd_add_expandable_type_D_aux:NN #1#2
1003 {
1004 \bool_if:NTF \l__cmd_long_bool
1005 { \cs_set:cpx }
1006 { \cs_set_nopar:cpx }
1007 { \l__cmd_expandable_aux_name_tl } ##1 #2 ##2 #2
1008 { ##1 {##2} }
1009 __cmd_add_expandable_grabber:nn { #1_alt }
1010 {
1011 \exp_not:c { \l__cmd_expandable_aux_name_tl }
1012 \exp_not:n {#2}
1013 }
1014 }

```

(End of definition for `\__cmd_add_expandable_type_D:w` and others.)

For each embellishment, use `\__cmd_get_grabber:NN` to obtain an auxiliary delimited by that token and store a pair constituted of the auxiliary and the token in `\l__cmd_tmpb_tl`, before appending the whole set of these pairs to the signature, and an equal number of `-NoValue-` markers (regardless of the default values of arguments). Set the current argument appropriately.

```

1015 \cs_new_protected:Npn __cmd_add_expandable_type_E:w #1#2
1016 {
1017 __cmd_add_default_E:nn {#1} {#2}
1018 \tl_clear:N \l__cmd_tmpb_tl
1019 \tl_map_function:nN {#1} __cmd_add_expandable_type_E_aux:n
1020 __cmd_add_expandable_grabber:nn
1021 { E \bool_if:NT \l__cmd_long_bool { _long } }
1022 {
1023 { \exp_not:o \l__cmd_tmpb_tl }
1024 }

```

```

1025 \prg_replicate:nn { \tl_count:n {#1} }
1026 { { \c_novalue_tl } }
1027 }
1028 }
1029 __cmd_prepare_signature:N
1030 }
1031 \cs_new_protected:Npn __cmd_add_expandable_type_E_aux:n #1
1032 {
1033 __cmd_get_grabber:NN #1 \l__cmd_tmpa_tl
1034 \tl_put_right:Nx \l__cmd_tmpb_tl
1035 { \exp_not:o \l__cmd_tmpa_tl \exp_not:N #1 }
1036 }

```

(End of definition for `\__cmd_add_expandable_type_E:w` and `\__cmd_add_expandable_type_E_aux:n`.)

`\__cmd_add_expandable_type_m:w` Unlike the standard case, when working expandably each argument is always grabbed separately.

```

1037 \cs_new_protected:Npn __cmd_add_expandable_type_m:w
1038 {
1039 __cmd_add_default:
1040 __cmd_add_expandable_grabber:nn
1041 { m \bool_if:NT \l__cmd_long_bool { _long } } { }
1042 __cmd_prepare_signature:N
1043 }

```

(End of definition for `\__cmd_add_expandable_type_m:w`.)

`\__cmd_add_expandable_type_R:w` The R-type is very similar to the D-type argument, and so the same internals are used.

```

1044 \cs_new_protected:Npn __cmd_add_expandable_type_R:w
1045 { __cmd_add_expandable_type_D_aux:NNNn R }

```

(End of definition for `\__cmd_add_expandable_type_R:w`.)

`\__cmd_add_expandable_type_t:w` An auxiliary delimited by #1 is built now. It will be used to test for the presence of that token.

```

1046 \cs_new_protected:Npn __cmd_add_expandable_type_t:w #1
1047 {
1048 __cmd_add_default:
1049 __cmd_get_grabber:NN #1 \l__cmd_tmpa_tl
1050 __cmd_add_expandable_grabber:nn { t }
1051 {
1052 \exp_not:o \l__cmd_tmpa_tl
1053 \exp_not:N #1
1054 }
1055 __cmd_prepare_signature:N
1056 }

```

(End of definition for `\__cmd_add_expandable_type_t:w`.)

`\__cmd_add_expandable_grabber:nn` This is called for all arguments to place the right grabber in the signature.

```

1057 \cs_new_protected:Npn __cmd_add_expandable_grabber:nn #1#2
1058 {
1059 \tl_put_right:Nx \l__cmd_signature_tl
1060 { \exp_not:c { __cmd_expandable_grab_ #1 :w } #2 }
1061 }

```

(End of definition for `\__cmd_add_expandable_grabber:nn`.)

```
__cmd_get_grabber:NN
__cmd_get_grabber_auxi:NN
__cmd_get_grabber_auxii:NN
```

Given a token #1, defines an expandable function delimited by that token and stores it in the token list #2. The function is named after the token, unless that function name is already taken by some other grabber (this can happen in the rare case where delimiters with different category codes are used in the same document): in that case use a global counter to get a unique name. Since the grabbers are not named after `xparse` commands they should not be used to get material from the input stream.

```
1062 \cs_new_protected:Npn __cmd_get_grabber:NN #1#2
1063 {
1064 \cs_set:Npn __cmd_tmp:w ##1 #1 {##1}
1065 \exp_args:Nc __cmd_get_grabber_auxi:NN
1066 { __cmd_grabber_ \token_to_str:N #1 :w } #2
1067 }
1068 \cs_new_protected:Npn __cmd_get_grabber_auxi:NN #1#2
1069 {
1070 \cs_if_eq:NNTF __cmd_tmp:w #1
1071 { \tl_set:Nn #2 {#1} }
1072 {
1073 \cs_if_exist:NTF #1
1074 {
1075 \int_gincr:N \g__cmd_grabber_int
1076 \exp_args:Nc __cmd_get_grabber_auxi:NN
1077 {
1078 __cmd_grabber_
1079 - \int_use:N \g__cmd_grabber_int :w
1080 }
1081 #2
1082 }
1083 { __cmd_get_grabber_auxii:NN #1 #2 }
1084 }
1085 }
1086 \cs_new_protected:Npn __cmd_get_grabber_auxii:NN #1#2
1087 {
1088 \cs_set_eq:NN #1 __cmd_tmp:w
1089 \tl_set:Nn #2 {#1}
1090 }
```

(End of definition for `\__cmd_get_grabber:NN`, `\__cmd_get_grabber_auxi:NN`, and `\__cmd_get_grabber_auxii:NN`.)

### 1.7.1 Copying a command and its internal structure

```
1091 ⟨!latexrelease⟩\IncludeInRelease{2021/11/15}{__cmd_copy:NN} %
1092 ⟨!latexrelease⟩ {Support~\NewCommandCopy-in-ltcmd}
```

Since the 2020-10-01 L<sup>A</sup>T<sub>E</sub>X 2<sub><</sub> release, support for copying, and showing the definition of, robust commands has been available, but the specifics of each command are implemented separately. Here we'll add support for copying and showing `ltcmd` definitions.

To fully support copying, we need two commands: a conditional to test if a command is in fact a `ltcmd` command, and another command to actually copy the command. The conditional is defined later as `\__kernel_cmd_if_xparse:NTF`, so now to the copying:

`\__cmd_copy:NN` This macro just branches to the proper copying command by using `\__cmd_cmd_type-cases:NnnnnnF`. The copying command takes the names of the commands to be copied to and from, and the actual commands as its four arguments.  
`\__cmd_set_eq_if_exist>NN`  
`\__cmd_set_eq_if_exist:cc`

```

1093 \cs_new_protected:Npn __cmd_copy:NN #1 #2
1094 {
1095 \use:x
1096 {
1097 \int_set:Nn \tex_escapechar:D { 92 }
1098 \exp_not:N __cmd_cmd_type_cases:NnnnnnF \exp_not:N #2
1099 { __cmd_copy_command:nnNN }
1100 { __cmd_copy_expandable:nnNN }
1101 { __cmd_copy_optimized:nnNN }
1102 { __cmd_copy_environment:nnNN }
1103 { __cmd_copy_environment_end:nnNN }
1104 { __cmd_cant_copy:nwn { non-ltcmd } }
1105 { \cs_to_str:N #1 } { \cs_to_str:N #2 }
1106 \exp_not:N #1 \exp_not:N #2
1107 \exp_not:N __cmd_break_point:n { \cs_to_str:N #2 }
1108 \int_set:Nn \tex_escapechar:D { \int_use:N \tex_escapechar:D }
1109 }
1110 }
1111 \cs_new_protected:Npn __cmd_set_eq_if_exist:NN #1 #2
1112 {
1113 \cs_if_exist:NTF #2 { \cs_set_eq:NN } { \use_none:nn } #1 #2
1114 \cs_generate_variant:Nn __cmd_set_eq_if_exist:NN { cc }

```

`\__cmd_cant_copy:nwn`  
An utility macro similar to `\__cmd_bad_def:wn` to abort a command copy. Contrary to `\__cmd_bad_def:wn` though, when this happens the issue is most likely internal, because the command was already (supposedly) correctly defined so it should be copyable. Hopefully this macro will never be used ever, but if it does, apologise and give the reason for the failure so the user can report.

```

1114 \cs_new_protected:Npn __cmd_cant_copy:nwn #1 #2 __cmd_break_point:n #3
1115 {
1116 \msg_error:nnnn { cmd } { copy-bug } { #1 } { #3 }
1117 {
1118 Error~while~copying~command~\iow_char:N\#2:\\
1119 \str_case:nn {#1}
1120 {
1121 { non-ltcmd } { Command~is~not~a~valid~ltcmd~command. }
1122 { unknown-type } { Found~an~unknown~argument~type. }
1123 { invalid-end }
1124 { Target~command~is~not~named~\iow_char:N \\end<name>. }
1125 }
1126 }

```

And, of course, add `\__kernel_cmd_if_xparse:NTF` and `\__cmd_copy:NN` to `\@declarecommandcopylisthook`:

```

1127 \tl_gput_right:Nn \@declarecommandcopylisthook
1128 { { __kernel_cmd_if_xparse:NTF __cmd_copy:NN } }

```

(End of definition for `\__cmd_copy:NN`, `\__cmd_set_eq_if_exist:NN`, and `\__cmd_cant_copy:nwn`.)

`\__cmd_copy_command:nnNN`  
`\__cmd_copy_command:NnNNnnnn`  
A normal (non-expandable) command has a pretty straightforward structure. Its definition is stored in `\⟨cmd⟩_code`, its defaults (if any) are stored in `\⟨cmd⟩_defaults`, and its

top-level definition contains its signature, which can just be copied over. `\__cmd_copy-command:nnNN` copies the command code and defaults, and then defines the top-level command using the auxiliary `\__cmd_copy_command:NnNNnnnn`. This macro takes the signature of the command being copied from its top-level definition, and replaces the named bits with the new name.

```

1129 \cs_new_protected:Npn __cmd_copy_command:nnNN #1 #2 #3 #4
1130 {
1131 \cs_set_eq:cc { #1 ~ code } { #2 ~ code }
1132 __cmd_set_eq_if_exist:cc { #1 ~ defaults } { #2 ~ defaults }
1133 \cs_set_protected_nopar:Npx #3
1134 { \exp_after:wN __cmd_copy_command:NnNNnnnn #4 {#1} }
1135 }
1136 \cs_new:Npn __cmd_copy_command:NnNNnnnn #1 #2 #3 #4 #5 #6 #7 #8
1137 {
1138 #1 \exp_not:n { {#2} }
1139 \exp_not:c { #8 ~ } \exp_not:c { #8 ~ code }
1140 \exp_not:n { {#5} {#6} {#7} }
1141 }
```

(End of definition for `\__cmd_copy_command:nnNN` and `\__cmd_copy_command:NnNNnnnn`.)

`\__cmd_copy_expandable:nnNN`  
`\__cmd_copy_expandable:NnNNNNnnnn`

An expandable command is slightly more complicated. Besides the `\langle cmd \rangle`, and `\langle cmd \rangle``defaults`, it also has an auxiliary `\langle cmd \rangle` for grabbing delimited arguments, and possibly another auxiliary `\langle cmd \rangle``ll`, if the command has both long and short arguments. Then, its signature also has several specific bits that are unique to that command; this is in contrast to non-expandable commands, which use a common set of parsing functions.

We start by copying the basics, then call `\__cmd_copy_expandable_signature:NnNNNNnnnn` to parse the signature of the command and build up the modified copy in a temporary token list, then we call `\__cmd_copy_expandable:NnNNNNnnnn` that will copy the top-level definition of the command, with the proper internal renames.

```

1142 ⟨latexrelease⟩\EndIncludeInRelease
1143 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{__cmd_copy:NN}%
1144 ⟨latexrelease⟩ {Support~\NewCommandCopy-in-ltcmd}
1145 ⟨latexrelease⟩\EndIncludeInRelease
```

There's one variant: a command begins with `\__cmd_start_expandable:nNNNNn` may still be un-expandable/protected if it's defined by `\NewDocumentCommand` and friends, with empty or only m-type arguments.

```

1146 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{__cmd_copy_expandable:nnNN}%
1147 ⟨latexrelease⟩ {Distinguish~non-expandable~document~commands}
1148 \cs_new_protected:Npn __cmd_copy_expandable:nnNN #1 #2 #3 #4
1149 {
1150 \cs_set_eq:cc { #1 ~ code } { #2 ~ code }
1151 __cmd_set_eq_if_exist:cc { #1 ~ } { #2 ~ }
1152 __cmd_set_eq_if_exist:cc { #1 ~ \c_space_t1 } { #2 ~ \c_space_t1 }
1153 __cmd_set_eq_if_exist:cc { #1 ~ defaults } { #2 ~ defaults }
1154 \exp_after:wN __cmd_copy_expandable_signature:NnNNNNnnnn #4 {#1} {#2}
1155 \token_if_protected_macro:NTF #4
1156 { \cs_set_protected_nopar:Npx } { \cs_set_nopar:Npx }
1157 #3
1158 { \exp_after:wN __cmd_copy_expandable:NnNNNNnnnn #4 {#1} {#2} }
1159 }
1160 ⟨latexrelease⟩\EndIncludeInRelease
```

```

1161 <|latexrelease>\IncludeInRelease{2021/11/15}{__cmd_copy_expandable:nnNN}%
1162 <|latexrelease> {Support~\NewCommandCopy~in~ltcmd}
1163 <|latexrelease>\cs_new_protected:Npn __cmd_copy_expandable:nnNN #1 #2 #3 #4
1164 <|latexrelease> {
1165 <|latexrelease> \cs_set_eq:cc { #1 ~ code } { #2 ~ code }
1166 <|latexrelease> __cmd_set_eq_if_exist:cc { #1 ~ } { #2 ~ }
1167 <|latexrelease> __cmd_set_eq_if_exist:cc { #1 ~ \c_space_tl } { #2 ~ \c_space_tl }
1168 <|latexrelease> __cmd_set_eq_if_exist:cc { #1 ~ defaults } { #2 ~ defaults }
1169 <|latexrelease> \exp_after:wN __cmd_copy_expandable_signature:NnNNNNnnn #4 {#1} {#2}
1170 <|latexrelease> \cs_set_nopar:Npx #3
1171 <|latexrelease> { \exp_after:wN __cmd_copy_expandable:NnNNNNnnn #4 {#1} {#2} }
1172 <|latexrelease>
1173 <|latexrelease>\EndIncludeInRelease
1174 <|latexrelease>\IncludeInRelease{2020/10/01}{__cmd_copy_expandable:nnNN}%
1175 <|latexrelease> {Support~\NewCommandCopy~in~ltcmd}
1176 <|latexrelease>\EndIncludeInRelease

```

Copy the code, simply define the wrapper.

```

1177 __cmd_copy_optimized:nnNN
1178 {
1179 <|cs_new_protected:Npn __cmd_copy_optimized:nnNN #1#2#3#4
1180 \token_if_protected_macro:NTF #4
1181 { \cs_set_protected_nopar:Npe }
1182 { \cs_set_nopar:Npe }
1183 #3
1184 {
1185 \exp_not:N __cmd_start_optimized:
1186 \exp_not:c { #1 ~ code }
1187 }
1188 }

1189 <|latexrelease>\IncludeInRelease{2021/11/15}{__cmd_copy:NN (part 2)}%
1190 <|latexrelease> {Support~\NewCommandCopy~in~ltcmd}
1191 \cs_new:Npn __cmd_copy_expandable:NnNNNNnnn #1 #2 #3 #4 #5 #6 #7 #8 #9
1192 {
1193 \exp_not:N #1 \exp_not:n { {#2} }
1194 \exp_not:c { #8 ~ }
1195 \exp_not:c
1196 {
1197 #8 ~
1198 \str_if_eq:eeT
1199 { \exp_not:c { #9 ~ \c_space_tl } } { \exp_not:N #4 }
1200 { \c_space_tl }
1201 }
1202 \exp_not:c { #8 ~ code }
1203 \str_if_eq:eeTF { \exp_not:N #6 } { ? }
1204 { ? }
1205 { \exp_not:c { #8 ~ defaults } }
1206 { \exp_not:V \l__cmd_tmpa_tl }
1207 }

```

A signature for an expandable command contains as many `\expandable_grab_{type}:w` as there are arguments, and what follows this macro depends on the `{type}`. We'll start a loop through the signature, and at each argument grabber, we'll step the argument

```

__cmd_copy_expandable_signature:NnNNNNnnn
__cmd_copy_expandable:nnN
__cmd_copy_parse_grabber:w

```

count, and look for the `<type>` with `\__cmd_copy_parse_grabber:w` so that we know which `\__cmd_copy_grabber_<type>:w` to call next.

```

1208 \cs_new_protected:Npn __cmd_copy_expandable_signature:NnNNNnnn
1209 #1 #2 #3 #4 #5 #6 #7 #8 #9
1210 {
1211 \int_zero:N \l__cmd_current_arg_int
1212 \tl_clear:N \l__cmd_tmpa_tl
1213 __cmd_copy_expandable:nnN {#8} {#9} #7
1214 \q_recursion_tail \q_recursion_stop
1215 }
1216 \cs_new_protected:Npn __cmd_copy_expandable:nnN #1 #2 #3
1217 {
1218 \quark_if_recursion_tail_stop:n {#3}
1219 \int_incr:N \l__cmd_current_arg_int
1220 \exp_after:wN __cmd_copy_parse_grabber:w \token_to_str:N #3 {#1} {#2}
1221 }
1222 \use:x
1223 {
1224 \cs_new_protected:Npn \exp_not:N __cmd_copy_parse_grabber:w ##1
1225 \tl_to_str:n { expandable_grab_ } ##2 \tl_to_str:n { :w }
1226 {
1227 \tl_put_right:Nx \exp_not:N \l__cmd_tmpa_tl
1228 { \exp_not:N \exp_not:c { __cmd_expandable_grab_##2:w } }
1229 \exp_not:N \cs_if_exist_use:cf { __cmd_copy_grabber_##2:w }
1230 { __cmd_cant_copy:nwn { unknown-type } }
1231 }
1232 }
1233 }
```

The most complicated is the Delimited argument: each argument has a dedicated grabbing function named after the command that has to be copied over (of the form `\(cmd)\u(arg\u(num))`).

```

1233 \cs_new_protected:Npn __cmd_copy_grabber_D:w #1 #2 #3 #4 #5
1234 {
1235 __cmd_copy_grabber_D:w
1236 __cmd_copy_grabber_D_alt:w
1237 __cmd_copy_grabber_R:w
1238 __cmd_copy_grabber_R_alt:w
1239 __cmd_copy_grabber_E:w
1240 __cmd_copy_grabber_E_long:w
1241 __cmd_copy_grabber_t:w
1242 __cmd_copy_grabber_m:w
1243 __cmd_copy_grabber_m_long:w
1244 }
```

`D_alt` is just a special case of `D` that uses a single delimiter (used when both delimiters of the argument are identical):

```

1245 \cs_new_protected:Npn __cmd_copy_grabber_D_alt:w #1 #2 #3 #4
1246 { __cmd_copy_grabber_D:w {#1} {#2} {#3} {#4} { } }
```

As far as copying is concerned, `R` is identical to `D`:

```

1247 \cs_new_eq:NN __cmd_copy_grabber_R:w __cmd_copy_grabber_D:w
1248 \cs_new_eq:NN __cmd_copy_grabber_R_alt:w __cmd_copy_grabber_D_alt:w
```

`E` is straightforward: we just copy the embellishments over, and increase the current argument number `\l__cmd_current_arg_int` by the number of embellishments (minus one because there is a `\int_incr:N` down the line).

```

1249 \cs_new_protected:Npn __cmd_copy_grabber_E:w #1 #2 #3 #4
1250 {
1251 \tl_put_right:Nn \l__cmd_tma_tl { [#3] [#4] }
1252 \int_add:Nn \l__cmd_current_arg_int { \tl_count:n [#4] - 1 }
1253 __cmd_copy_expandable:nnN [#1] [#2]
1254 }
1255 \cs_new_eq:NN __cmd_copy_grabber_E_long:w __cmd_copy_grabber_E:w
t just needs copying the token to be tested for:
1256 \cs_new_protected:Npn __cmd_copy_grabber_t:w #1 #2 #3 #4
1257 {
1258 \tl_put_right:Nn \l__cmd_tma_tl { #3 #4 }
1259 __cmd_copy_expandable:nnN [#1] [#2]
1260 }
```

And last but not least, `m` is the simplest; the grabber is just `\__cmd_expandable_grab_m:w`, which is already added to the new command so here we just resume the loop:

```

1261 \cs_new_protected:Npn __cmd_copy_grabber_m:w { __cmd_copy_expandable:nnN }
1262 \cs_new_eq:NN __cmd_copy_grabber_m_long:w __cmd_copy_grabber_m:w
```

(End of definition for `\__cmd_copy_expandable:nnNN` and others.)

`\__cmd_copy_environment:nnNN`  
`\__cmd_copy_environment:Nnnnnnnn`

Copying an environment's `\begin` part is pretty much like copying a command, except it has a longer name, and at the end we have to copy `\environment <name>` into `\<name>`.

```

1263 \cs_new_protected:Npn __cmd_copy_environment:nnNN #1 #2 #3 #4
1264 {
1265 \cs_set_eq:cc { environment~ #1 ~ code } { environment~ #2 ~ code }
1266 __cmd_set_eq_if_exist:cc
1267 { environment~ #1 ~ defaults } { environment~ #2 ~ defaults }
1268 \cs_set_protected_nopar:cpx { environment~ #1 }
1269 { \exp_after:wN __cmd_copy_environment:Nnnnnnn #4 [#1] }
1270 \cs_set_eq:cc [#1] { environment~ #1 }
1271 }
1272 \cs_new:Npn __cmd_copy_environment:Nnnnnnnn #1 #2 #3 #4 #5 #6 #7
1273 { #1 \exp_not:n { [#2] } { #7} \exp_not:n { [#4] [#5] [#6] } }
```

(End of definition for `\__cmd_copy_environment:nnNN` and `\__cmd_copy_environment:Nnnnnnnn`.)

`\__cmd_copy_environment_end:nnNN`  
`\__cmd_copy_environment_end_aux:nnNN`

Copying an environment's `\end` part is a bit trickier. We first have to make sure that both parts are named `\end<name>` (that's actually not a hard requirement, but an environment `\end` command makes no sense without the `end` in its name), and strip the leading `end` from the strings. After that, copying is straightforward.

```

1274 \cs_new_protected:Npn __cmd_copy_environment_end:nnNN #1 #2
1275 {
1276 __cmd_check_end:Nn \l__cmd_tma_tl [#1]
1277 __cmd_check_end:Nn \l__cmd_tmpb_t1 [#2]
1278 \exp_args:Noo __cmd_copy_environment_end_aux:nnNN
1279 { \l__cmd_tma_tl } { \l__cmd_tmpb_t1 }
1280 }
1281 \cs_new_protected:Npn __cmd_copy_environment_end_aux:nnNN #1 #2 #3 #4
1282 {
```

```

1283 \cs_set_nopar:cpx { environment~ #1 ~end }
1284 { \exp_not:c { environment~ #1 ~end-aux } }
1285 \cs_set_eq:cc
1286 { environment~ #1 ~end-aux~ } { environment~ #2 ~end-aux~ }
1287 \cs_set_eq:cc { end #1 } { environment~ #1 ~end }
1288 }

```

To check whether an `\end` command is valid, we look for the string `end` at the beginning of the command name, and if not found, raise an error:

```

__cmd_check_end:Nn
__cmd_check_end:n
__cmd_check_end:w
1289 \cs_new_protected:Npn __cmd_check_end:Nn #1 #2
1290 {
1291 \tl_set:Nx #1 { __cmd_check_end:n {#2} }
1292 \token_if_eq_meaning:NNT #1 \q_nil
1293 { __cmd_cant_copy:nwn { invalid-end } }
1294 }
1295 \cs_set_protected:Npn __cmd_tmp:w #1
1296 {
1297 \cs_new:Npn __cmd_check_end:n ##1
1298 {
1299 \exp_after:wN __cmd_check_end:w \tl_to_str:n {##1}
1300 #1 \q_mark #1 \q_stop
1301 }
1302 \cs_new:Npn __cmd_check_end:w ##1 #1 ##2 #1 ##3 \q_stop
1303 { \if_meaning:w ##2 \q_mark \exp_not:N \q_nil \else: ##2 \fi: }
1304 }
1305 \exp_args:No __cmd_tmp:w { \tl_to_str:n { end } }

(End of definition for __cmd_copy_environment_end:nnNN and others.)

```

Not much to do regarding `\texrelease`: we could remove the entries from `\@declarecommandcopylist` but it doesn't seem worth it.

```

1306 <\texrelease> \EndIncludeInRelease
1307 <\texrelease> \IncludeInRelease{2020/10/01}{__cmd_copy:NN (part 2)}%
1308 <\texrelease> {Support~\NewCommandCopy-in-ltcmd}
1309 <\texrelease> \EndIncludeInRelease

```

### 1.7.2 Showing the definition of a command

```

1310 <\texrelease> \IncludeInRelease{2021/11/15}{__cmd_show:N}%
1311 <\texrelease> {Support~\ShowCommand-in-ltcmd}

```

To show the definition of a command we need more or less the same building blocks as for copying, except that instead of making a copy, we'll just print stuff to the terminal. This macro just branches to the proper showing command by using `\__cmd_cmd_type_cases:NnnnnnnF`. The showing command takes the command to be shown as argument.

```

1312 \cs_new_protected:Npn __cmd_show:N #1
1313 {
1314 \use:x
1315 {
1316 \int_set:Nn \tex_escapechar:D { 92 }
1317 \exp_not:N __cmd_cmd_type_cases:NnnnnnnF \exp_not:N #1
1318 { __cmd_show_command:N }
1319 { __cmd_show_expandable:N }
1320 { __cmd_show_optimized:N }
1321 { __cmd_show_environment:N }

```

```

1322 { __cmd_show_environment_end:N }
1323 { __cmd_cant_copy:nwn { non-ltcmd } }
1324 \exp_not:N #1
1325 \exp_not:N __cmd_break_point:n { \cs_to_str:N #1 }
1326 \int_set:Nn \tex_escapechar:D { \int_use:N \tex_escapechar:D }
1327 }
1328 }
```

(End of definition for `\__cmd_show:N`.)

These commands just expand the command once to reveal its innards, then pass the type of command, the control sequence, the signature, and the code macro to `\__cmd_show_command_aux:NnNNn`.

```

1329 \cs_new_protected:Npn __cmd_show_command:N #1
1330 { \exp_after:wN __cmd_show_command:NnNNwN #1 \q__cmd #1 }
1331 \cs_new_protected:Npn __cmd_show_command:NnNNwN #1 #2 #3 #4 #5 \q__cmd #6
1332 {
1333 __cmd_show_command_aux:NnNNn \tl_show:x
1334 { document~command } #6 #4 {#2}
1335 }
1336 \cs_new_protected:Npn __cmd_show_expandable:N #1
1337 { \exp_after:wN __cmd_show_expandable:NnNNNNnN #1 #1 }
1338 \end{IncludeInRelease}
1339 \IncludeInRelease{2020/10/01}{__cmd_show:N}%
1340 \Support~\ShowCommand~in~ltcmd}
1341 \end{IncludeInRelease}
```

There's one variant: a command begins with `\__cmd_start_expandable:nNNNNn` may still be un-expandable/protected if it's defined by `\NewDocumentCommand` and friends, with empty or only m-type arguments.

```

1342 \end{IncludeInRelease}
1343 \begin{IncludeInRelease}{2023/06/01}{__cmd_show_expandable:NnNNNNnN}%
1344 \Support~non-expandable-document~commands}
1345 \cs_new_protected:Npn __cmd_show_expandable:NnNNNNnN #1 #2 #3 #4 #5 #6 #7 #8
1346 {
1347 \exp_args:NNe __cmd_show_command_aux:NnNNn \tl_show:x
1348 { \token_if_protected_macro:NF #8 { expandable~ } document~command }
1349 #8 #5 {#2}
1350 }
1351 \end{IncludeInRelease}
1352 \begin{IncludeInRelease}{2021/11/15}{__cmd_show_expandable:NnNNNNnN}%
1353 \Support~\ShowCommand~in~ltcmd}
1354 \cs_new_protected:Npn __cmd_show_expandable:NnNNNNnN #1 #2 #3 #4 #5 #6 #7 #8
1355 {
1356 __cmd_show_command_aux:NnNNn \tl_show:x
1357 { expandable~document~command } #8 #5 {#2}
1358 }
1359 \end{IncludeInRelease}
1360 \begin{IncludeInRelease}{2020/10/01}{__cmd_show_expandable:NnNNNNnN}%
1361 \Support~\ShowCommand~in~ltcmd}
1362 \begin{IncludeInRelease}{2021/11/15}{__cmd_show:N (part 2)}%
1363 \Support~\ShowCommand~in~ltcmd}
```

Now just print everything in the required format. The auxiliary `\__cmd_split_signature:n` stores a ready-to-print token list in `\l__cmd_tmpa_tl`, so we just use that here:

```

1364 \cs_new_protected:Npn __cmd_show_command_aux:NnNNn #1 #2 #3 #4 #5
1365 {
1366 __cmd_split_signature:n {#5}
1367 #1
1368 {
1369 \token_to_str:N #3 = #2: \iow_newline:
1370 \tl_use:N \l__cmd_tmpa_tl
1371 -> \cs_replacement_spec:N #4
1372 }
1373 }
```

Optimized functions need things done a bit differently as we need to reconstruct the argument spec.

```

1374 \cs_new_protected:Npn __cmd_show_optimized:N #1
1375 {
1376 \exp_args:Nc __cmd_show_optimized:NN
1377 { \cs_to_str:N #1 \c_space_tl code }
1378 #1
1379 }
1380 \cs_new_protected:Npn __cmd_show_optimized:NN #1#2
1381 {
1382 \cs_set:Npe __cmd_show_optimized_aux:N ##1
1383 {
1384 \c_space_tl \c_space_tl \c_hash_str ##1 :
1385 \bool_lazy_or:nnT
1386 { \token_if_long_macro_p:N #1 }
1387 { \token_if_protected_long_macro_p:N #1 }
1388 { + } m
1389 \iow_newline:
1390 }
1391 \tl_show:e
1392 {
1393 \token_to_str:N #2 =
1394 \bool_lazy_or:nnF
1395 { \token_if_protected_macro_p:N #1 }
1396 { \token_if_protected_long_macro_p:N #1 }
1397 { expandable ~ } document~command:
1398 \iow_newline:
1399 \int_step_function:nN
1400 {
1401 \int_div_truncate:nn
1402 { \tl_count:e { \cs_parameter_spec:N #1 } }
1403 { 2 }
1404 }
1405 __cmd_show_optimized_aux:N
1406 ->
1407 \cs_replacement_spec:N #1
1408 }
1409 }
1410 \cs_generate_variant:Nn \tl_count:n { e }
```

We can reuse most of the above to show an environment, except that we need to ensure that the proper `\environment` ... are passed to `\_cmd_show_command_-aux:NnNNn`. Additionally, when `\ShowCommand\foo` is used (if `foo` is an environment), we show `\endfoo` as well, and when `\ShowCommand\endfoo` is used, change that to `\ShowCommand\foo` and do the same.

```

1411 \cs_new_protected:Npn _cmd_show_environment:N #1
1412 {
1413 \exp_after:wN _cmd_show_environment:Nnnw #1 \q_ cmd
1414 \tl_show:x
1415 {
1416 \token_to_str:N \end { \cs_to_str:N #1 } : \iow_newline:
1417 -> \exp_args:Nc \cs_replacement_spec:N
1418 { environment~ \cs_to_str:N #1 ~end~aux~ }
1419 }
1420 }
1421 \cs_new_protected:Npn _cmd_show_environment:Nnnw #1 #2 #3 #4 \q_ cmd
1422 {
1423 \use:x
1424 {
1425 _cmd_show_command_aux:NnNNn _cmd_show:x { document~environment }
1426 { \exp_not:N \begin {#3} }
1427 \exp_not:c { environment~ #3 ~ code }
1428 {#2}
1429 }
1430 }
1431 \cs_new_protected:Npn _cmd_show:x #1
1432 { \iow_term:x { > ~ #1 . \iow_newline: } }
1433 \cs_new_protected:Npn _cmd_show_environment_end:N #1
1434 {
1435 \exp_args:NNx _cmd_check_end:Nn \l_ cmd_tpa_t1 { \cs_to_str:N #1 }
1436 \exp_args:Nc _cmd_show_environment:N { \l_ cmd_tpa_t1 }
1437 }
```

And, of course, add `\_kernel_cmd_if_xparse:NTF` and `\_cmd_show:N` to `\@showcommandlisthook` and to `\@showenvironmentlisthook` (`\_cmd_show:N` takes care of the environment case as well, so both entries are identical):

```

1438 \tl_gput_right:Nn \@showcommandlisthook
1439 { { _kernel_cmd_if_xparse:NTF _cmd_show:N } }
1440 \tl_gput_right:Nn \@showenvironmentlisthook
1441 { { _kernel_cmd_if_xparse:NTF _cmd_show:N } }
```

*(End of definition for `\_cmd_show_command:N` and others.)*

`\_cmd_split_signature:n` Now we'll try a least-effort adventure into splitting the symbolic user-provided signature for a command into individual parameters for pretty-printing. A counter is used to keep track of the current argument number, and two token lists are used: `\l_ cmd_tpa_t1` holds the final token list to be printed, and `\l_ cmd_tmpb_t1` holds just the current item, so that we can make changes to an individual item without having to dissect the whole thing (this is used for e- and E-types).

```

1442 \cs_new_protected:Npn _cmd_split_signature:n #1
1443 {
1444 \int_set:Nn \l_ cmd_current_arg_int { 1 }
1445 \tl_clear:N \l_ cmd_tpa_t1
```

```

1446 \tl_clear:N \l__cmd_tmpb_tl
1447 __cmd_split_signature_loop:Nw #1 \q_recursion_tail \q_recursion_stop
1448 }

```

This is the main chunk of the loop: it starts an item with `\__cmd_split_start_item:` (this adds indentation and the argument number to `\l__cmd_tmpb_tl`), then checks if a special token list `\c__cmd_show_type_<type>_tl` exists. If it doesn't, the current argument is a “simple” type which needs no extra processing. Otherwise, call a specific function depending on the value of said token list.

```

1449 \cs_new_protected:Npn __cmd_split_signature_loop:Nw #1
1450 {
1451 \quark_if_recursion_tail_stop:N #1
1452 \tl_if_empty:NT \l__cmd_tmpb_tl { __cmd_split_start_item: }
1453 \tl_if_exist:cTF { \c__cmd_show_type_#1_tl }
1454 {
1455 \use:c
1456 {
1457 __cmd_show_
1458 \if_case:w \tl_use:c { \c__cmd_show_type_#1_tl } \exp_stop_f:
1459 delim \or: delims \or: delims_opt \or: opt \or:
1460 e \or: E \or: prefix \or: processor \fi: :Nw
1461 } #1
1462 }
1463 { __cmd_split_end_item:n {#1} __cmd_split_signature_loop:Nw }
1464 }

```

The token lists `\c__cmd_show_type_<type>_tl` exist for nontrivial (for printing) `<types>` that require special parsing (like delimiters or optional arguments). Values from 0 to 7 are assigned to each type:

1. a single delimiter token;
2. two delimiter tokens;
3. two delimiter tokens plus a default value;
4. a default value;
5. a list of embellishments (exclusive for `e`-type);
6. embellishments plus defaults (exclusive for `E`-type);
7. simple prefixes;
8. prefixes with arguments (argument processors);

```

1465 \cs_set_protected:Npn __cmd_tmp:w #1 #2
1466 {
1467 \quark_if_nil:nF {#1}
1468 { \tl_const:cn { \c__cmd_show_type_#1_tl } {#2} __cmd_tmp:w }
1469 }
1470 __cmd_tmp:w t0 r1 d1 R2 D2 03 e4 E5 +6 !6 >7 =7 \q_nil \q_nil

```

Now, based on each type we know how to act. In most cases it is just a matter of feeding in the grabbed arguments and resuming the loop. The embellishments require a bit more attention: the e-type loops through the list of embellishments and adds each to the token list as a separate argument. The E-type does more or less the same, but uses `\__cmd_tl_mapthread_function:nnN` to map over two lists simultaneously, adding each token and default to the token list for printing.

```

1471 \cs_new_protected:Npn __cmd_show_delim:Nw #1 #2
1472 { __cmd_split_end_item:n { #1 #2 } __cmd_split_signature_loop:Nw }
1473 \cs_new_protected:Npn __cmd_show_delims:Nw #1 #2 #3
1474 { __cmd_split_end_item:n { #1 #2 #3 } __cmd_split_signature_loop:Nw }
1475 \cs_new_protected:Npn __cmd_show_delims_opt:Nw #1 #2 #3 #4
1476 { __cmd_split_end_item:n { #1 #2 #3 {#4} } __cmd_split_signature_loop:Nw }
1477 \cs_new_protected:Npn __cmd_show_opt:Nw #1 #2
1478 { __cmd_split_end_item:n { #1 {#2} } __cmd_split_signature_loop:Nw }
1479 \cs_new_protected:Npn __cmd_show_e:Nw #1 #2
1480 {
1481 \tl_map_inline:nn {#2}
1482 {
1483 __cmd_split_start_item:
1484 __cmd_split_end_item:n { #1 ##1 }
1485 }
1486 __cmd_split_signature_loop:Nw
1487 }
1488 \cs_set_protected:Npn __cmd_tmp:w #1
1489 {
1490 \cs_new_protected:Npn __cmd_show_E:Nw ##1 ##2 ##3
1491 {
1492 \cs_set_protected:Npn __cmd_tmp:w #####1 #####2
1493 {
1494 __cmd_split_start_item:
1495 __cmd_split_end_item:n { ##1 #####1 #####2 }
1496 }
1497 __cmd_tl_mapthread_function:nnN {##2}
1498 { ##3 {#1} {#1} {#1} {#1} {#1} {#1} {#1} {#1} } __cmd_tmp:w
1499 __cmd_split_signature_loop:Nw
1500 }
1501 }
1502 \exp_args:NV __cmd_tmp:w \c_novalue_tl

```

Minor wrinkle with the prefixes: they use `\__cmd_split_add_item:n` instead of `\__cmd_split_end_item:n` (add *vs.* end) because they are followed by an argument, so they can't end the item.

```

1503 \cs_new_protected:Npn __cmd_show_prefix:Nw #
1504 { __cmd_split_add_item:n {#1} __cmd_split_signature_loop:Nw }
1505 \cs_new_protected:Npn __cmd_show_processor:Nw #1 #2
1506 { __cmd_split_add_item:n { #1 {#2} } __cmd_split_signature_loop:Nw }

```

And now the auxiliaries that store the strings to be printed. `\__cmd_split_start_item:` starts an item from scratch, `\__cmd_split_add_item:n` adds tokens to an item without adding a newline, and `\__cmd_split_end_item:n` adds tokens, terminates the item with a newline, and steps the argument count.

```

1507 \cs_new_protected:Npn __cmd_split_start_item:
1508 {

```

```
1509 \tl_set:Nx \l__cmd_tmpb_tl
1510 { ~ \c_space_tl \c_hash_str \int_use:N \l__cmd_current_arg_int : }
1511 }
1512 \cs_new_protected:Npn __cmd_split_add_item:n #1
1513 { \tl_put_right:Nx \l__cmd_tmpb_tl { \tl_to_str:n {#1} } }
1514 \cs_new_protected:Npn __cmd_split_end_item:n #1
1515 {
1516 \tl_put_right:Nx \l__cmd_tmptl
1517 { \l__cmd_tmpb_tl \tl_to_str:n {#1} \iow_newline: }
1518 \tl_clear:N \l__cmd_tmpb_tl
1519 \int_incr:N \l__cmd_current_arg_int
1520 }
```

(End of definition for `\_cmd\_split\_signature:n` and others.)

Not much to do regarding `latexrelease`: we could remove the entries from `\@showcommandlisthook`, but it doesn't seem worth it.

```
1521 \end{IncludeInRelease}
1522 %
1523 \IncludeInRelease{2020/10/01}{_cmd_show:N (part 2)}%
1524 {\Support~\ShowCommand~in~\ltcmd}
1525 \end{IncludeInRelease}
```

## 1.8 Grabbing arguments

All of the grabbers follow the same basic pattern. The initial function stores in \lCmdSignatureTl the code to grab further arguments, defines (the function in) \lCmdFnTl that will grab the argument, and calls it.

Defining `\l1__cmd_fn_t1` means determining whether to use `\cs_set:Npn` or `\cs_set_nopar:Npn`, and for optional arguments whether to skip spaces. Once the argument is found, `\l1__cmd_fn_t1` calls `\__cmd_add_arg:n`, responsible for calling processors and grabbing further arguments.

This uses the well-tested code of D-type arguments, skipping the peeking step because the b-type argument is always present, and adding a cleanup stage at the end by hijacking the signature. The clean-up consists of properly dealing with `\l__cmd_args_tl` and also putting back the `\end` that served as an end-delimiter: this `\end` receives the environment name as its argument and is run normally. The D-type code stores the argument found (body of the environment) as a brace group in `\l__cmd_args_tl` and depending on the presence of a prefix ! we trim spaces or not before adding this braced argument into the saved `\l__cmd_args_tl`. The strange `\begin_` control sequence is there for display purposes only: it has to look like `\begin` in the terminal but not to delimited arguments.

```
1526 \cs_new_protected:Npn __cmd_grab_b:w
1527 { __cmd_grab_b_aux:NNw \cs_set_protected_nopar:Npn \tl_trim_spaces:n }
1528 \cs_new_protected:Npn __cmd_grab_b_long:w
1529 { __cmd_grab_b_aux:NNw \cs_set_protected:Npn \tl_trim_spaces:n }
1530 \cs_new_protected:Npn __cmd_grab_b_obey_spaces:w
1531 { __cmd_grab_b_aux:NNw \cs_set_protected_nopar:Npn \exp_not:n }
1532 \cs_new_protected:Npn __cmd_grab_b_long_obey_spaces:w
1533 { __cmd_grab_b_aux:NNw \cs_set_protected:Npn \exp_not:n }
1534 \cs_new_protected:Npn __cmd_grab_b_aux:NNw #1#2#3 __cmd_run_code:
1535 {
1536 __cmd_grab_D_aux>NNnNN \begin \end {#3} #1 \use_i:nn
```

```

1537 \tl_put_left:Nn \l__cmd_signature_tl { __cmd_grab_b_end:Nw #2 }
1538 \tl_set_eq:NN \l__cmd_saved_args_tl \l__cmd_args_tl
1539 \tl_clear:N \l__cmd_args_tl
1540 \exp_args:Nc \l__cmd_fn_tl { begin ~ }
1541 }
1542 \cs_new_protected:Npn __cmd_grab_b_end:Nw #1#2 __cmd_run_code:
1543 {
1544 \tl_set:Nx \l__cmd_args_tl
1545 {
1546 \exp_not:V \l__cmd_saved_args_tl
1547 { \exp_after:wN #1 \l__cmd_args_tl }
1548 }
1549 #2
1550 __cmd_run_code:
1551 \end
1552 }

```

(End of definition for `\__cmd_grab_b:w` and others.)

`\__cmd_grab_D:w`

`\__cmd_grab_D_long:w`

`\__cmd_grab_D_obey_spaces:w`

`\__cmd_grab_D_long_obey_spaces:w`

`\__cmd_grab_D_no_strip:w`

`\__cmd_grab_D_long_no_strip:w`

`\__cmd_grab_D_obey_spaces_no_strip:w`

`\__cmd_grab_D_long_obey_spaces_no_strip:w`

The generic delimited argument grabber. The auxiliary function does a peek test before calling `\__cmd_grab_D_call:Nw`, so that the optional nature of the argument works as expected.

```

1553 \cs_new_protected:Npn __cmd_grab_D:w #1#2#3 __cmd_run_code:
1554 {
1555 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected_nopar:Npn
1556 __cmd_peek_nonspace_remove:NTF \use_i:nn
1557 }
1558 \cs_new_protected:Npn __cmd_grab_D_long:w #1#2#3 __cmd_run_code:
1559 {
1560 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected:Npn
1561 __cmd_peek_nonspace_remove:NTF \use_i:nn
1562 }
1563 \cs_new_protected:Npn __cmd_grab_D_obey_spaces:w #1#2#3 __cmd_run_code:
1564 {
1565 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected_nopar:Npn
1566 __cmd_peek_meaning_remove:NTF \use_i:nn
1567 }
1568 \cs_new_protected:Npn __cmd_grab_D_long_obey_spaces:w #1#2#3 __cmd_run_code:
1569 {
1570 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected:Npn
1571 __cmd_peek_meaning_remove:NTF \use_i:nn
1572 }
1573 \cs_new_protected:Npn __cmd_grab_D_no_strip:w
1574 #1#2#3 __cmd_run_code:
1575 {
1576 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected_nopar:Npn
1577 __cmd_peek_nonspace_remove:NTF \use_none:n
1578 }
1579 \cs_new_protected:Npn __cmd_grab_D_long_no_strip:w
1580 #1#2#3 __cmd_run_code:
1581 {
1582 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected:Npn
1583 __cmd_peek_nonspace_remove:NTF \use_none:n
1584 }

```

```

1585 \cs_new_protected:Npn __cmd_grab_D_obey_spaces_no_strip:w
1586 #1#2#3 __cmd_run_code:
1587 {
1588 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected_nopar:Npn
1589 __cmd_peek_meaning_remove:NTF \use_none:n
1590 }
1591 \cs_new_protected:Npn __cmd_grab_D_long_obey_spaces_no_strip:w
1592 #1#2#3 __cmd_run_code:
1593 {
1594 __cmd_grab_D_aux:NNnNNN #1 #2 {#3} \cs_set_protected:Npn
1595 __cmd_peek_meaning_remove:NTF \use_none:n
1596 }

```

This is a bit complicated. The idea is that, in order to check for nested optional argument tokens ([[[...]]] and so on) the argument needs to be grabbed without removing any braces at all. If this is not done, then cases like [{ [ ]}] fail. So after testing for an optional argument, it is collected piece-wise. Inserting a quark prevents loss of braces, and there is then a test to see if there are nested delimiters to handle.

```

1597 \cs_new_protected:Npn __cmd_grab_D_aux:NNnNNN #1#2#3#4#5#6
1598 {
1599 __cmd_grab_D_aux:NNnNNN #1#2 {#3} #4 #6
1600 #5 #1
1601 { __cmd_grab_D_call:Nw #1 }
1602 { __cmd_add_arg:o \c_novalue_tl }
1603 }

```

Inside the “standard” grabber, there is a test to see if the grabbed argument is entirely enclosed by braces. There are a couple of extra factors to allow for: the argument might be entirely empty, and spaces at the start and end of the input must be retained around a brace group. Also notice that a *blank* argument might still contain spaces. To allow for suppression of brace stripping, the business end is passed here as #5.

```

1604 \cs_new_protected:Npn __cmd_grab_D_aux:NNnNNN #1#2#3#4#5
1605 {
1606 \tl_set:Nn \l__cmd_signature_tl {#3}
1607 \exp_after:wN #4 \l__cmd_fn_tl ##1 #2
1608 {
1609 \tl_if_in:nnTF {##1} {#1}
1610 { __cmd_grab_D_nested:NNnN #1 #2 {##1} #4 }
1611 {
1612 \tl_if_blank:oTF { \use_none:n ##1 }
1613 { __cmd_add_arg:o { \use_none:n ##1 } }
1614 {
1615 \str_if_eq:eeTF
1616 { \exp_not:o { \use_none:n ##1 } }
1617 { { \exp_not:o { \use_i:nnn ##1 \q_nil } } }
1618 { __cmd_add_arg:o { #5 ##1 } }
1619 { __cmd_add_arg:o { \use_none:n ##1 } }
1620 }
1621 }
1622 }
1623 }

```

(End of definition for \\_\_cmd\_grab\_D:w and others.)

```
__cmd_grab_D_nested:NNnN
 __cmd_grab_D_nested:w
 \l_cmd_nesting_a_tl
 \l_cmd_nesting_b_tl
 \q_cmd
```

Catching nested optional arguments means more work. The aim here is to collect up each pair of optional tokens without TeX helping out, and without counting anything. The code above will already have removed the leading opening token and a closing token, but the wrong one. The aim is then to work through the material grabbed so far and divide it up on each opening token, grabbing a closing token to match (thus working in pairs). Once there are no opening tokens, then there is a second check to see if there are any opening tokens in the second part of the argument (for things like [] []). Once everything has been found, the entire collected material is added to the output as a single argument. The only tricky part here is ensuring that any grabbing function that might run away is named after the function currently being parsed and not after `xparse`. That leads to some rather complex nesting! There is also a need to prevent the loss of any braces, hence the insertion and removal of quarks along the way.

```
1624 \tl_new:N \l_cmd_nesting_a_tl
1625 \tl_new:N \l_cmd_nesting_b_tl
1626 \quark_new:N \q_cmd
1627 \cs_new_protected:Npn __cmd_grab_D_nested:NNnN #1#2#3#4
 {
 \tl_clear:N \l_cmd_nesting_a_tl
 \tl_clear:N \l_cmd_nesting_b_tl
 \exp_after:wN #4 \l_cmd_fn_tl ##1 #1 ##2 \q_cmd ##3 #2
 {
 \tl_put_right:No \l_cmd_nesting_a_tl { \use_none:n ##1 #1 }
 \tl_put_right:No \l_cmd_nesting_b_tl { \use_i:nn #2 ##3 }
 \tl_if_in:nnTF {##2} {#1}
 {
 \l_cmd_fn_tl
 \q_nil ##2 \q_cmd \ERROR
 }
 {
 \tl_put_right:Nx \l_cmd_nesting_a_tl
 { __cmd_grab_D_nested:w \q_nil ##2 \q_stop }
 \tl_if_in:NnTF \l_cmd_nesting_b_tl {#1}
 {
 \tl_set_eq:NN \l_cmd_tmpa_tl \l_cmd_nesting_b_tl
 \tl_clear:N \l_cmd_nesting_b_tl
 \exp_after:wN \l_cmd_fn_tl \exp_after:wN
 \q_nil \l_cmd_tmpa_tl \q_nil \q_cmd \ERROR
 }
 {
 \tl_put_right:No \l_cmd_nesting_a_tl
 \l_cmd_nesting_b_tl
 __cmd_add_arg:V \l_cmd_nesting_a_tl
 }
 }
 }
 \l_cmd_fn_tl #3 \q_nil \q_cmd \ERROR
}
\cs_new:Npn __cmd_grab_D_nested:w #1 \q_nil \q_stop
 { \exp_not:o { \use_none:n #1 } }
```

(End of definition for `\_\_cmd_grab_D_nested:NNnN` and others.)

```
__cmd_grab_D_call:Nw
```

For D and R-type arguments, to avoid losing any braces, a token needs to be inserted before the argument to be grabbed. If the argument runs away because the closing token

is missing then this inserted token shows up in the terminal. Ideally, #1 would therefore be used directly, but that is no good as it will mess up the rest of the grabber. Instead, a copy of #1 with an altered category code is used, as this will look right in the terminal but will not mess up the grabber. The only issue then is that the category code of #1 is unknown. So there is a quick test to ensure that the inserted token can never be matched by the grabber. (This assumes that the open and close delimiters are not the same character with different category codes, but that really should not happen in any sensible document-level syntax.) An exception is when #1 is a control sequence token, in which case the character-token treatment is no good because if hit with \token\_to\_-str:N it would add spurious tokens to the argument. In this case a different branch is taken. The token inserted is then the same  $\langle csname \rangle$  as #1, but with a space appended, so that the grabber don't see it as another of the same delimiter.

```

1661 \cs_new_protected_nopar:Npn __cmd_grab_D_call:Nw #1
1662 {
1663 \token_if_eq_catcode:NNTF + #1
1664 {
1665 \exp_after:wN \exp_after:wN \exp_after:wN
1666 \l__cmd_fn_tl \char_generate:nn { '#1 } { 11 }
1667 }
1668 {
1669 __cmd_token_if_cs:NTF #1
1670 {
1671 \exp_after:wN \l__cmd_fn_tl
1672 \cs:w \cs_to_str:N #1 ~ \cs_end:
1673 }
1674 {
1675 \exp_after:wN \l__cmd_fn_tl
1676 \token_to_str:N #1
1677 }
1678 }
1679 }
```

(End of definition for \\_\_cmd\_grab\_D\_call:Nw.)

\\_\_cmd\_grab\_E:w Everything here needs to point to a loop.

```

__cmd_grab_E_long:w 1680 \cs_new_protected:Npn __cmd_grab_E:w #1#2 __cmd_run_code:
__cmd_grab_E_obey_spaces:w 1681 {
__cmd_grab_E_long_obey_spaces:w 1682 __cmd_grab_E:nnNN {#1} {#2}
__cmd_grab_E:nnNN 1683 \cs_set_protected_nopar:Npn
__cmd_grab_E_loop:NnN 1684 __cmd_peek_nonspace_remove:NTF
__cmd_grab_E_finalise: 1685 }
1686 \cs_new_protected:Npn __cmd_grab_E_long:w #1#2 __cmd_run_code:
1687 {
1688 __cmd_grab_E:nnNN {#1} {#2}
1689 \cs_set_protected:Npn
1690 __cmd_peek_nonspace_remove:NTF
1691 }
1692 \cs_new_protected:Npn __cmd_grab_E_obey_spaces:w #1#2 __cmd_run_code:
1693 {
1694 __cmd_grab_E:nnNN {#1} {#2}
1695 \cs_set_protected_nopar:Npn
1696 __cmd_peek_meaning_remove:NTF
1697 }
```

```

1698 \cs_new_protected:Npn __cmd_grab_E_long_obey_spaces:w #1#2 __cmd_run_code:
1699 {
1700 __cmd_grab_E:nnNN {#1} {#2}
1701 \cs_set_protected:Npn
1702 __cmd_peek_meaning_remove:NTF
1703 }

```

A loop is needed here to allow a random ordering of keys. These are searched for one at a time, with any not found needing to be tracked: they can appear later. The grabbed values are held in a property list which is then turned into an ordered list to be passed back to the user.

```

1704 \cs_new_protected:Npn __cmd_grab_E:nnNN #1#2#3#4
1705 {
1706 \exp_after:wN #3 \l__cmd_fn_tl ##1##2##3
1707 {
1708 \prop_put:Nnn \l__cmd_tmp_prop {##1} {##3}
1709 __cmd_grab_E_loop:NnN #4 { } ##2 \q_recursion_stop
1710 }
1711 \prop_clear:N \l__cmd_tmp_prop
1712 \tl_set:Nn \l__cmd_signature_tl {#2}
1713 \cs_set_protected:Npn __cmd_grab_E_finalise:
1714 {
1715 \tl_map_inline:nn {#1}
1716 {
1717 \prop_get:NnNF \l__cmd_tmp_prop {####1} \l__cmd_tmpb_tl
1718 { \tl_set_eq:NN \l__cmd_tmpb_tl \c_novalue_tl }
1719 \tl_put_right:Nx \l__cmd_args_tl
1720 { { \exp_not:V \l__cmd_tmpb_tl } }
1721 }
1722 \l__cmd_signature_tl __cmd_run_code:
1723 }
1724 __cmd_grab_E_loop:NnN #4 { } #1 \q_recursion_tail \q_recursion_stop
1725 }
1726 \cs_new_protected:Npn __cmd_grab_E_loop:NnN #1#2#3#4 \q_recursion_stop
1727 {
1728 \cs_if_eq:NNTF #3 \q_recursion_tail
1729 { __cmd_grab_E_finalise: }
1730 {
1731 #1 #3
1732 { \l__cmd_fn_tl #3 {#2#4} }
1733 { __cmd_grab_E_loop:NnN #1 {#2#3} #4 \q_recursion_stop }
1734 }
1735 }
1736 \cs_new_protected:Npn __cmd_grab_E_finalise: { }

(End of definition for __cmd_grab_E:w and others.)

```

\\_\_cmd\_grab\_m:w  
\\_\_cmd\_grab\_m\_long:w Collecting a single mandatory argument is quite easy.

```

1737 \cs_new_protected:Npn __cmd_grab_m:w #1 __cmd_run_code:
1738 {
1739 \tl_set:Nn \l__cmd_signature_tl {#1}
1740 \exp_after:wN \cs_set_protected_nopar:Npn \l__cmd_fn_tl ##1
1741 { __cmd_add_arg:n {##1} }
1742 \l__cmd_fn_tl

```

```

1743 }
1744 \cs_new_protected:Npn __cmd_grab_m_long:w #1 __cmd_run_code:
1745 {
1746 \tl_set:Nn \l__cmd_signature_tl {#1}
1747 \exp_after:wN \cs_set_protected:Npn \l__cmd_fn_tl ##1
1748 { __cmd_add_arg:n {##1} }
1749 \l__cmd_fn_tl
1750 }
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790

(End of definition for __cmd_grab_m:w and __cmd_grab_m_long:w.)

__cmd_grab_m_1:w Grabbing 1–8 mandatory arguments is done by giving 8–1 known arguments to a 9-
__cmd_grab_m_2:w argument function that stores them in \l__cmd_args_tl. For simplicity, grabbing 9
__cmd_grab_m_3:w mandatory arguments is done by grabbing 5 then 4 arguments.
__cmd_grab_m_4:w
__cmd_grab_m_5:w
__cmd_grab_m_6:w
__cmd_grab_m_7:w
__cmd_grab_m_8:w
__cmd_grab_m_9:w
__cmd_grab_m_aux:Nnnnnnnnnn \cs_new_protected_nopar:Npn __cmd_grab_m_aux:Nnnnnnnnnn #1#2#3#4#5#6#7#8#9
{
 \tl_put_right:No \l__cmd_args_tl
 { #1 {#2} {#3} {#4} {#5} {#6} {#7} {#8} {#9} }
 \l__cmd_signature_tl __cmd_run_code:
}
\cs_new_protected:cpn { __cmd_grab_m_1:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
 \l__cmd_fn_tl \use_none:nnnnnnn { } { } { } { } { } { } { }
}
\cs_new_protected:cpn { __cmd_grab_m_2:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
 \l__cmd_fn_tl \use_none:nnnnnnn { } { } { } { } { } { }
}
\cs_new_protected:cpn { __cmd_grab_m_3:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
 \l__cmd_fn_tl \use_none:nnnnn { } { } { } { } { }
}
\cs_new_protected:cpn { __cmd_grab_m_4:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
 \l__cmd_fn_tl \use_none:nnnnn { } { } { } { }
}
\cs_new_protected:cpn { __cmd_grab_m_5:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
 \l__cmd_fn_tl \use_none:nnn { } { } { }
}
\cs_new_protected:cpn { __cmd_grab_m_6:w } #1 __cmd_run_code:
{
 \tl_set:Nn \l__cmd_signature_tl {#1}
 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_tl __cmd_grab_m_aux:Nnnnnnnnn
}

```

```

1791 \l__cmd_fn_t1 \use_none:nn { } { }
1792 }
1793 \cs_new_protected:cpn { __cmd_grab_m_7:w } #1 __cmd_run_code:
1794 {
1795 \tl_set:Nn \l__cmd_signature_t1 {#1}
1796 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_t1 __cmd_grab_m_aux:Nnnnnnnnn
1797 \l__cmd_fn_t1 \use_none:n { }
1798 }
1799 \cs_new_protected:cpn { __cmd_grab_m_8:w } #1 __cmd_run_code:
1800 {
1801 \tl_set:Nn \l__cmd_signature_t1 {#1}
1802 \exp_after:wN \cs_set_eq:NN \l__cmd_fn_t1 __cmd_grab_m_aux:Nnnnnnnnn
1803 \l__cmd_fn_t1 \prg_do_nothing:
1804 }
1805 \cs_new_protected:cpx { __cmd_grab_m_9:w }
1806 {
1807 \exp_not:c { __cmd_grab_m_5:w }
1808 \exp_not:c { __cmd_grab_m_4:w }
1809 }

```

(End of definition for `\__cmd_grab_m_1:w` and others.)

`\__cmd_grab_R:w`  
`\__cmd_grab_R_long:w`  
`\__cmd_grab_R_aux:NNnN`

```

1810 \cs_new_protected:Npn __cmd_grab_R:w #1#2#3 __cmd_run_code:
1811 { __cmd_grab_R_aux:NNnN #1 #2 {#3} \cs_set_protected_nopar:Npn }
1812 \cs_new_protected:Npn __cmd_grab_R_long:w #1#2#3 __cmd_run_code:
1813 { __cmd_grab_R_aux:NNnN #1 #2 {#3} \cs_set_protected:Npn }
1814 \cs_new_protected:Npn __cmd_grab_R_aux:NNnN #1#2#3#4
1815 {
1816 __cmd_grab_D_aux:NNnNN #1 #2 {#3} #4 \use_ii:nn
1817 __cmd_peek_nonspace_remove:NTF #1
1818 { __cmd_grab_D_call:Nw #1 }
1819 {
1820 \msg_error:nxxx { cmd } { missing-required }
1821 { __cmd_environment_or_command: }
1822 { \token_to_str:N #1 }
1823 __cmd_add_arg:o \c_novalue_t1
1824 }
1825 }

```

(End of definition for `\__cmd_grab_R:w`, `\__cmd_grab_R_long:w`, and `\__cmd_grab_R_aux:NNnN`.)

`\__cmd_grab_t:w`  
`\__cmd_grab_t_obey_spaces:w`  
`\__cmd_grab_t_aux>NNw`

```

1826 \cs_new_protected:Npn __cmd_grab_t:w
1827 { __cmd_grab_t_aux>NNw __cmd_peek_nonspace_remove:NTF }
1828 \cs_new_protected:Npn __cmd_grab_t_obey_spaces:w
1829 { __cmd_grab_t_aux>NNw __cmd_peek_meaning_remove:NTF }
1830 \cs_new_protected:Npn __cmd_grab_t_aux>NNw #1#2#3 __cmd_run_code:
1831 {
1832 \tl_set:Nn \l__cmd_signature_t1 {#3}
1833 \exp_after:wN \cs_set_protected:Npn \l__cmd_fn_t1
1834 {

```

```

1835 #1 #2
1836 { __cmd_add_arg:n { \BooleanTrue } }
1837 { __cmd_add_arg:n { \BooleanFalse } }
1838 }
1839 \l__cmd_fn_tl
1840 }

```

(End of definition for `\__cmd_grab_t:w`, `\__cmd_grab_t_obey_spaces:w`, and `\__cmd_grab_t_aux:NNw`.)

`\l__cmd_v_arg_tl` <sub>1841</sub> `\tl_new:N \l__cmd_v_arg_tl`

`\__cmd_grab_v:w` Firstly, it is necessary to change `\tex_endlinechar:D` so that newlines in different catcode regimes (e.g., `\ExplSyntaxOn`) are not misinterpreted as spaces. The opening delimiter is the first non-space token, and is never read verbatim. This is required by consistency with the case where the preceding argument was optional and absent: then TeX has already read and tokenized that token when looking for the optional argument. The first thing is thus to check is that this delimiter is a character, and to distinguish the case of a left brace (in that case, `\group_align_safe_end:` is needed to compensate for the begin-group character that was just seen). Then set verbatim catcodes with `\__cmd_grab_v_aux_catcodes::`.

The group keep catcode changes local, and `\group_align_safe_begin/end:` allow to use a character with category code 4 (normally `&`) as the delimiter (all commands do `\group_align_safe_begin/end:`, so there's no need to do that again here). It is ended by `\__cmd_grab_v_group_end:`, which smuggles the collected argument out of the group.

```

1842 \cs_new_protected:Npn __cmd_grab_v:w
1843 {
1844 \bool_set_false:N \l__cmd_long_bool
1845 __cmd_grab_v_aux:w
1846 }
1847 \cs_new_protected:Npn __cmd_grab_v_long:w
1848 {
1849 \bool_set_true:N \l__cmd_long_bool
1850 __cmd_grab_v_aux:w
1851 }
1852 \cs_new_protected:Npn __cmd_grab_v_aux:w #1 __cmd_run_code:
1853 {
1854 \tl_set:Nn \l__cmd_signature_tl {\#1}
1855 \group_begin:
1856 \tex_escapechar:D = 92 \scan_stop:
1857 \tex_endlinechar:D = '\^M \scan_stop:
1858 \tl_clear:N \l__cmd_v_arg_tl
1859 \peek_remove_spaces:n
1860 {
1861 \peek_meaning_remove:NTF \c_group_begin_token
1862 {
1863 \group_align_safe_end:
1864 __cmd_grab_v_bgroup:
1865 }
1866 {
1867 \peek_N_type:TF

```

```

1868 { __cmd_grab_v_aux_test:N }
1869 { __cmd_grab_v_aux_abort:n { } }
1870 }
1871 }
1872 }
1873 \cs_new_protected:Npn __cmd_grab_v_group_end:
1874 {
1875 \exp_args:NNNo
1876 \group_end:
1877 \tl_set:Nn \l__cmd_v_arg_tl { \l__cmd_v_arg_tl }
1878 }

```

(End of definition for `\__cmd_grab_v:w` and others.)

```

__cmd_grab_v_aux_test:N
__cmd_grab_v_aux_loop:N
__cmd_grab_v_aux_loop>NN
__cmd_grab_v_aux_loop_end:

```

Check that the opening delimiter is a character, setup category codes, then start reading tokens one by one, keeping the delimiter as an argument. If the verbatim was not nested, we will be grabbing one character at each step. Unfortunately, it can happen that what follows the verbatim argument is already tokenized. Thus, we check at each step that the next token is indeed a “nice” character, *i.e.*, is not a character with category code 1 (begin-group), 2 (end-group) or 6 (macro parameter), nor the space character, with category code 10 and character code 32, nor a control sequence. The partially built argument is stored in `\l__cmd_v_arg_tl`. If we ever meet a token which we cannot grab (non-N-type), or which is not a character according to `\__cmd_grab_v_token_if_char:NTF`, then we bail out with `\__cmd_grab_v_aux_abort:n`. Otherwise, we stop at the first character matching the delimiter.

```

1879 \cs_new_protected:Npn __cmd_grab_v_aux_test:N #1
1880 {
1881 __cmd_grab_v_token_if_char:NTF #1
1882 {
1883 __cmd_grab_v_aux_put:N #1
1884 __cmd_grab_v_aux_catcodes:
1885 __cmd_grab_v_aux_loop:N #1
1886 }
1887 { __cmd_grab_v_aux_abort:n {#1} #1 }
1888 }
1889 \cs_new_protected:Npn __cmd_grab_v_aux_loop:N #1
1890 {
1891 \peek_N_type:TF
1892 { __cmd_grab_v_aux_loop>NN #1 }
1893 { __cmd_grab_v_aux_abort:n { } }
1894 }
1895 \cs_new_protected:Npn __cmd_grab_v_aux_loop>NN #1#2
1896 {
1897 __cmd_grab_v_token_if_char:NTF #2
1898 {
1899 \token_if_eq_charcode:NNTF #1 #2
1900 { __cmd_grab_v_aux_loop_end: }
1901 {
1902 __cmd_grab_v_aux_put:N #2
1903 __cmd_grab_v_aux_loop:N #1
1904 }
1905 }
1906 { __cmd_grab_v_aux_abort:n {#2} #2 }

```

```

1907 }
1908 \cs_new_protected:Npn __cmd_grab_v_aux_loop_end:
1909 {
1910 __cmd_grab_v_group_end:
1911 __cmd_add_arg:x { \tl_tail:N \l__cmd_v_arg_tl }
1912 }

```

(End of definition for `\__cmd_grab_v_aux_test:N` and others.)

`\l__cmd_v_nesting_int` 1913 `\int_new:N \l__cmd_v_nesting_int`

`\__cmd_grab_v_bgroup:` If the opening delimiter is a left brace, we keep track of how many left and right braces were encountered so far in `\l__cmd_v_nesting_int` (the methods used for optional arguments cannot apply here), and stop as soon as it reaches 0.

`\__cmd_grab_v_bgroup_loop:N` Some care was needed when removing the opening delimiter, which has already been assigned category code 1: using `\peek_meaning_remove:NTF` in the `\__cmd_grab_v_aux:w` function would break within alignments. Instead, we first convert that token to a string, and remove the result as a normal undelimited argument.

```

1914 \cs_new_protected:Npx __cmd_grab_v_bgroup:
1915 {
1916 \exp_not:N __cmd_grab_v_aux_catcodes:
1917 \exp_not:n { \int_set:Nn \l__cmd_v_nesting_int { 1 } }
1918 \exp_not:N __cmd_grab_v_aux_put:N \iow_char:N \{
1919 \exp_not:N __cmd_grab_v_bgroup_loop:
1920 }
1921 \cs_new_protected:Npn __cmd_grab_v_bgroup_loop:
1922 {
1923 \peek_N_type:TF
1924 { __cmd_grab_v_bgroup_loop:N }
1925 { __cmd_grab_v_aux_abort:n { } }
1926 }
1927 \cs_new_protected:Npn __cmd_grab_v_bgroup_loop:N #1
1928 {
1929 __cmd_grab_v_token_if_char:NTF #1
1930 {
1931 \token_if_eq_charcode:NNTF \c_group_end_token #1
1932 {
1933 \int_decr:N \l__cmd_v_nesting_int
1934 \int_compare:nNnTF \l__cmd_v_nesting_int > 0
1935 {
1936 __cmd_grab_v_aux_put:N #1
1937 __cmd_grab_v_bgroup_loop:
1938 }
1939 { __cmd_grab_v_aux_loop_end: }
1940 }
1941 }
1942 \token_if_eq_charcode:NNT \c_group_begin_token #1
1943 { \int_incr:N \l__cmd_v_nesting_int }
1944 __cmd_grab_v_aux_put:N #1
1945 __cmd_grab_v_bgroup_loop:
1946 }
1947 }

```

```

1948 { __cmd_grab_v_aux_abort:n {#1} #1 }
1949 }

(End of definition for __cmd_grab_v_bgroup:, __cmd_grab_v_bgroup_loop:, and
__cmd_grab_v_bgroup_loop:N.)
```

\\_\_cmd\_grab\_v\_aux\_catcodes: The approach for short verbatim arguments is to make the end-line character a macro parameter character: this is forbidden by the rest of the code. Then the error branch can check what caused the bail out and give the appropriate error message.

```

1950 \cs_new_protected:Npn __cmd_grab_v_aux_catcodes:
1951 {
1952 \cs_set_eq:NN \do \char_set_catcode_other:N
1953 \dospecials
1954 \bool_if:NTF \l__cmd_long_bool
1955 { \char_set_catcode_other:n { \tex_endlinechar:D } }
1956 { \char_set_catcode_parameter:n { \tex_endlinechar:D } }
1957 }
1958 \cs_new_protected:Npn __cmd_grab_v_aux_abort:n #1
1959 {
1960 __cmd_grab_v_group_end:
1961 \exp_after:wN \exp_after:wN \exp_after:wN
1962 \peek_meaning_remove:NTF \char_generate:nn { \tex_endlinechar:D } { 6 }
1963 {
1964 \msg_error:nnxxx { cmd } { verbatim-nl }
1965 { __cmd_environment_or_command: }
1966 { \tl_to_str:N \l__cmd_v_arg_tl }
1967 { \tl_to_str:n {#1} }
1968 __cmd_add_arg:o \c_novalue_tl
1969 }
1970 {
1971 \msg_error:nnxxx { cmd } { verbatim-tokenized }
1972 { __cmd_environment_or_command: }
1973 { \tl_to_str:N \l__cmd_v_arg_tl }
1974 { \tl_to_str:n {#1} }
1975 __cmd_add_arg:o \c_novalue_tl
1976 }
1977 }
```

(End of definition for \\_\_cmd\_grab\_v\_aux\_catcodes: and \\_\_cmd\_grab\_v\_aux\_abort:n.)

\\_\_cmd\_grab\_v\_aux\_put:N Storing one token in the collected argument. Most tokens are converted to category code 12, with the exception of active characters, and spaces (not sure what should be done for those).

```

1978 \langle latexrelease \rangle \IncludeInRelease{2024/06/01}{__cmd_grab_v_aux_put:N}%
1979 \langle latexrelease \rangle {Endlines~as~\obeyedline}
1980 \cs_new_protected:Npn __cmd_grab_v_aux_put:N #1
1981 {
1982 \tl_put_right:Nx \l__cmd_v_arg_tl
1983 {
1984 \token_if_active:NTF #1
1985 { \exp_not:N #1 }
1986 {
1987 \int_compare:nNnTF {'#1} = \tex_endlinechar:D
1988 { \exp_not:N \obeyedline }
```

```

1989 { \token_to_str:N #1 }
1990 }
1991 }
1992 }
1993 <|latexrelease> \EndIncludeInRelease
1994 <|latexrelease> \IncludeInRelease{2020/10/01}{__cmd_grab_v_aux_put:N}%
1995 <|latexrelease> {Endlines~as~\obeyedline}
1996 <|latexrelease> \cs_new_protected:Npn __cmd_grab_v_aux_put:N #1
1997 <|latexrelease> {
1998 <|latexrelease> \tl_put_right:Nx \l__cmd_v_arg_tl
1999 <|latexrelease> {
2000 <|latexrelease> \token_if_active:NTF #1
2001 <|latexrelease> { \exp_not:N #1 } { \token_to_str:N #1 }
2002 <|latexrelease> }
2003 <|latexrelease> }
2004 <|latexrelease> \EndIncludeInRelease

(End of definition for __cmd_grab_v_aux_put:N.)

```

\\_\_cmd\_grab\_v\_token\_if\_char:NTF

This function assumes that the escape character is printable. Then the string representation of control sequences is at least two characters, and \str\_tail:n only removes the escape character. Macro parameter characters are doubled by \tl\_to\_str:n, and will also yield a non-empty result, hence are not considered as characters.

```

2005 \cs_new_protected:Npn __cmd_grab_v_token_if_char:NTF #1
2006 { \str_if_eq:eeTF { } { \str_tail:n {#1} } }

```

(End of definition for \\_\_cmd\_grab\_v\_token\_if\_char:NTF.)

```

__cmd_add_arg:n When an argument is found it is stored, then further arguments are grabbed by calling
__cmd_add_arg:V \l__cmd_signature_tl.
__cmd_add_arg:o
__cmd_add_arg:x
2007 \cs_new_protected:Npn __cmd_add_arg:n #1
2008 {
2009 \tl_put_right:Nn \l__cmd_args_tl { {#1} }
2010 \l__cmd_signature_tl __cmd_run_code:
2011 }
2012 \cs_generate_variant:Nn __cmd_add_arg:n { V , o , x }

(End of definition for __cmd_add_arg:n.)

```

## 1.9 Grabbing arguments expandably

The first step is to grab the first token or group. The generic grabbers \(\langle function \rangle\_{\sqcup}\) and \(\langle function \rangle\_{\sqcap}\) are just after \q\_cmd, we go and find them (and use the long one).

```

2013 \cs_new:Npn __cmd_expandable_grab_D:w #1 \q_cmd #2#3
2014 { #2 { __cmd_expandable_grab_D:NNNwNNNn #1 \q_cmd #2 #3 } }

```

We then wish to test whether #7, which we just grabbed, is exactly #2. A preliminary test is whether their string representations coincide, then expand the only grabber function we have, #1, once: the two strings below are equal if and only if #7 matches #2 exactly.<sup>2</sup> The

---

<sup>2</sup>It is obvious that if #7 matches #2 then the strings are equal. We must check the converse. The right-hand-side of \str\_if\_eq:onTF does not end with #3, implying that the grabber function took everything as its arguments. The first brace group can only be empty if #7 starts with #2, otherwise the brace group preceding #7 would not vanish. The third brace group is empty, thus the \q\_cmd that was used by our grabber #1 must be the one that we inserted (not some token in #7), hence the second brace group contains the end of #7 followed by #2. Since this is #2 on the right-hand-side, and no brace can be lost there, #7 must contain nothing else than its leading #2.

preliminary test is needed as #7 could validly contain \par (because a later mandatory argument could be long) and our grabber may be short. If #7 does not match #2, then the optional argument is missing, we use the default `-NoValue-`, and put back the argument #7 in the input stream.

If it does match, then interesting things need to be done. We will grab the argument piece by piece, with the following pattern:

```
<grabber> {<tokens>}
\q_nil {<piece 1>} <piece 2> \ERROR \q__cmd
\q_nil <input stream>
```

The `<grabber>` will find an opening delimiter in `<piece 2>`, take the `\q__cmd` as a second delimiter, and find more material delimited by the closing delimiter in the `<input stream>`. We then move the part before the opening delimiter from `<piece 2>` to `<piece 1>`, and the material taken from the `<input stream>` to the `<piece 2>`. Thus, the argument moves gradually from the `<input stream>` to the `<piece 2>`, then to the `<piece 1>` when we have made sure to find all opening and closing delimiters. This two-step process ensures that nesting works: the number of opening delimiters minus closing delimiters in `<piece 1>` is always equal to the number of closing delimiters in `<piece 2>`. We stop grabbing arguments once the `<piece 2>` contains no opening delimiter any more, hence the balance is reached, and the final argument is `<piece 1> <piece 2>`. The indirection via `\__cmd_tmp:w` allows to insert `-NoValue-` expanded.

```
2015 \cs_set_protected:Npn __cmd_tmp:w #1
2016 {
2017 \cs_new:Npn __cmd_expandable_grab_D:NNNwNNn ##1##2##3##4 \q__cmd ##5##6##7
2018 {
2019 \str_if_eq:nnTF {##2} {##7}
2020 {
2021 \str_if_eq:onTF
2022 { ##1 { } { } ##7 ##2 \q__cmd ##3 }
2023 { { } {##2} { } }
2024 }
2025 { \use_i:nn }
2026 {
2027 ##1
2028 { __cmd_expandable_grab_D:NNNwNNnnn ##1##2##3##4 \q__cmd ##5##6 }
2029 \q_nil { } ##2 \ERROR \q__cmd \ERROR
2030 }
2031 { ##4 {#1} \q__cmd ##5 ##6 {##7} }
2032 }
2033 }
2034 \exp_args:N __cmd_tmp:w { \c_novalue_t1 }
```

At this stage, #7 is `\q_nil {<piece 1>} <more for piece 1>`, and we want to concatenate all that, removing `\q_nil`, and keeping the opening delimiter #2. Simply use `\use_i:nn`. Also, #8 is `<remainder of piece 2> \ERROR`, and #9 is `\ERROR <more for piece 2>`. We concatenate those, replacing the two `\ERROR` by the closing delimiter #3.

```
2035 \cs_new:Npn __cmd_expandable_grab_D:NNNwNNnnn #1#2#3#4 \q__cmd #5#6#7#8#9
2036 {
2037 \exp_args:Nof __cmd_expandable_grab_D:nnNNNwNN
2038 { \use_i:nn #7 #2 }
2039 { __cmd_expandable_grab_D:Nw #3 \exp_stop_f: #8 #9 }
2040 #1#2#3 #4 \q__cmd #5 #6
```

```

2041 }
2042 \cs_new:Npn __cmd_expandable_grab_D:Nw #1#2 \ERROR \ERROR { #2 #1 }

Armed with our two new ⟨pieces⟩, we are ready to loop. However, we must first see if ⟨piece 2⟩ (here #2) contains any opening delimiter #4. Again, we expand #3, this time removing its whole output with \use_none:nnn. The test is similar to \tl_if_in:nnTF. The token list is empty if and only if #2 does not contain the opening delimiter. In that case, we are done, and put the argument (from which we remove a spurious pair of delimiters coming from how we started the loop). Otherwise, we go back to looping with __cmd_expandable_grab_D:NNNwNNnnn. The code to deal with brace stripping is much the same as for the non-expandable case.

2043 \cs_new:Npn __cmd_expandable_grab_D:nnNNNwNN #1#2#3#4#5#6 \q__cmd #7#8
2044 {
2045 \exp_args:No \tl_if_empty:TF
2046 { #3 { \use_none:nnn } #2 \q__cmd #5 #4 \q__cmd #5 }
2047 {
2048 \tl_if_blank:oTF { \use_none:nn #1#2 }
2049 { __cmd_put_arg_expandable:ow { \use_none:nn #1#2 } }
2050 {
2051 \str_if_eq:eeTF
2052 { \exp_not:o { \use_none:nn #1#2 } }
2053 { { \exp_not:o { \use_iii:nnnn #1#2 \q_nil } } }
2054 { __cmd_put_arg_expandable:ow { \use_iii:nnn #1#2 } }
2055 { __cmd_put_arg_expandable:ow { \use_none:nn #1#2 } }
2056 }
2057 #6 \q__cmd #7 #8
2058 }
2059 {
2060 #3
2061 { __cmd_expandable_grab_D:NNNwNNnnn #3#4#5#6 \q__cmd #7 #8 }
2062 \q_nil {#1} #2 \ERROR \q__cmd \ERROR
2063 }
2064 }

```

(End of definition for \\_\_cmd\_expandable\_grab\_D:w and others.)

\\_\_cmd\_expandable\_grab\_D:alt:  
\\_\_cmd\_expandable\_grab\_D:alt:NNwNNn  
\\_\_cmd\_expandable\_grab\_D:alt:Nwn

```

2065 \cs_new:Npn __cmd_expandable_grab_D_alt:w #1 \q__cmd #2#3
2066 { #2 { __cmd_expandable_grab_D_alt:NNwNNn #1 \q__cmd #2 #3 } }
2067 \cs_set_protected:Npn __cmd_tmp:w #
2068 {
2069 \cs_new:Npn __cmd_expandable_grab_D_alt:NNwNNn ##1##2##3 \q__cmd ##4##5##6
2070 {
2071 \str_if_eq:nnTF {##6} {##2}
2072 {
2073 \str_if_eq:onTF
2074 { ##1 { } ##6 ##2 ##2 }
2075 { { } ##2 }
2076 }
2077 { \use_ii:nn }
2078 {
2079 ##1

```

```

2080 { __cmd_expandable_grab_D_alt>NNwn ##4 ##5 ##3 \q__cmd }
2081 ##6 \ERROR
2082 }
2083 { ##3 {#1} \q__cmd ##4 ##5 {##6} }
2084 }
2085 }
2086 \exp_args:No __cmd_tmp:w { \c_novalue_t1 }
2087 \cs_new:Npn __cmd_expandable_grab_D_alt>NNwn #1#2#3 \q__cmd #4
2088 {
2089 \tl_if_blank:oTF { \use_none:n #4 }
2090 { __cmd_put_arg_expandable:ow { \use_none:n #4 } }
2091 {
2092 \str_if_eq:eeTF
2093 { \exp_not:o { \use_none:n #4 } }
2094 { { \exp_not:o { \use_ii:nnn #4 \q_nil } } }
2095 { __cmd_put_arg_expandable:ow { \use_ii:nn #4 } }
2096 { __cmd_put_arg_expandable:ow { \use_none:n #4 } }
2097 }
2098 #3 \q__cmd #1 #2
2099 }

```

(End of definition for `\__cmd_expandable_grab_D_alt:w`, `\__cmd_expandable_grab_D_alt>NNwNNn`, and `\__cmd_expandable_grab_D_alt:Nwn`.)

We keep track of long/short by placing the appropriate grabber as the third token after `\q__cmd`; it is eventually removed by the `end:nnw` auxiliary. The `aux:w` auxiliary will be called repeatedly with two arguments: the set of pairs `<parser> <token>`, and the set of arguments found so far (initially all `{-NoValue-}`). At each step, grab what follows in the input stream then call the `loop:nnnNNw` auxiliary to compare it with each possible embellishment in turn. This auxiliary's #1 is what was found in the input, #2 collects `<parser> <token>` pairs that did not match, #3 collects the corresponding arguments found previously, #4 and #5 is the current pair, #6 is the remaining pairs, #7 is empty or two `\q_nil`, and #8 is the current argument. If none of the pairs matched (determined by `\quark_if_nil:NTF`) then call the `end` auxiliary to stop looking for embellishments, remembering to put what was grabbed in the input back where it belongs, and storing the arguments found just before `\q__cmd`. If the current argument #8 is not `-NoValue-` or if the input #1 does not match #5 (see t-type arguments below for a similar `\str_if_eq:onTF` test) then carry on the loop. Otherwise, we found a new embellishment: grab the corresponding argument in the input using the `find:w` auxiliary. To avoid losing braces around that auxiliary's argument we include a space, which will be eliminated in the next loop through embellishments.

```

2100 \cs_new:Npn __cmd_expandable_grab_E:w #1 \q__cmd #2#3
2101 { __cmd_expandable_grab_E_aux:w #1 \q__cmd #2 #3 #3 }
2102 \cs_new:Npn __cmd_expandable_grab_E_long:w #1 \q__cmd #2#3
2103 { __cmd_expandable_grab_E_aux:w #1 \q__cmd #2 #3 #2 }
2104 \cs_new:Npn __cmd_expandable_grab_E_aux:w #1 \q__cmd #2#3#4
2105 { #2 { __cmd_expandable_grab_E_test:nnw #1 \q__cmd #2 #3 #4 } }
2106 \cs_new:Npn __cmd_expandable_grab_E_test:nnw #1#2#3 \q__cmd #4#5#6#7
2107 {
2108 __cmd_expandable_grab_E_loop:nnnNNw {#7} { } { }
2109 #1 \q_nil \q_nil \q_nil \q_mark #2 \q_nil
2110 #3 \q__cmd #4 #5 #6
2111 }

```

```

2112 \cs_new:Npn __cmd_expandable_grab_E_loop:nnnNNw
2113 #1#2#3#4#5#6 \q_nil #7 \q_mark #8
2114 {
2115 \quark_if_nil:NTF #4
2116 { __cmd_expandable_grab_E_end:nnw {#1} {#3} }
2117 {
2118 \tl_if_novalue:nTF {#8}
2119 { \str_if_eq:onTF {#4} { } #1 #5 } {#5} }
2120 { \use_i:nn }
2121 { __cmd_expandable_grab_E_find:w {#2} {#4} {#5} {#6} } {#3} ~ }
2122 {
2123 __cmd_expandable_grab_E_loop:nnnNNw
2124 {#1} {#2} {#4} {#5} {#3} {#8} }
2125 #6 \q_nil #7 \q_mark
2126 }
2127 }
2128 }
2129 \cs_new:Npn __cmd_expandable_grab_E_find:w #1 \q_cmd #2#3#4
2130 { #4 { __cmd_expandable_grab_E_find:nnw #1 \q_cmd #2 #3 #4 } }
2131 \cs_new:Npn __cmd_expandable_grab_E_find:nnw #1#2#3 \q_nil #4 \q_cmd #5#6#7#8
2132 { __cmd_expandable_grab_E_aux:w {#1} {#2} {#8} {#3} #4 \q_cmd #5 #6 #7 }
2133 \cs_new:Npn __cmd_expandable_grab_E_end:nnw #1#2#3 \q_cmd #4#5#6
2134 { #3 #2 \q_cmd #4 #5 {#1} }

```

(End of definition for `\__cmd_expandable_grab_E:w` and others.)

```
__cmd_expandable_grab_m:w
 \cmd_expandable_grab_m_long:w
 \cmd_expandable_grab_m_aux:wNn
```

The mandatory case is easy: find the auxiliary after the `\q_cmd`, and use it directly to grab the argument, then correctly position the argument before `\q_cmd`.

```

2135 \cs_new:Npn __cmd_expandable_grab_m:w #1 \q_cmd #2#3
2136 { #3 { __cmd_expandable_grab_m_aux:wNn #1 \q_cmd #2 #3 } }
2137 \cs_new:Npn __cmd_expandable_grab_m_long:w #1 \q_cmd #2#3
2138 { #2 { __cmd_expandable_grab_m_aux:wNn #1 \q_cmd #2 #3 } }
2139 \cs_new:Npn __cmd_expandable_grab_m_aux:wNn #1 \q_cmd #2#3#4
2140 { #1 {#4} \q_cmd #2 #3 }
```

(End of definition for `\__cmd_expandable_grab_m:w`, `\__cmd_expandable_grab_m_long:w`, and `\__cmd_expandable_grab_m_aux:wNn`)

```
__cmd_expandable_grab_R:w
 \cmd_expandable_grab_R_aux:NNNwNNn
```

Much the same as for the D-type argument, with only the lead-off function varying.

```

2141 \cs_new:Npn __cmd_expandable_grab_R:w #1 \q_cmd #2#3
2142 { #2 { __cmd_expandable_grab_R_aux:NNNwNNn #1 \q_cmd #2#3 } }
2143 \cs_set_protected:Npn __cmd_tmp:w #1
2144 {
2145 \cs_new:Npn __cmd_expandable_grab_R_aux:NNNwNNn ##1##2##3##4 \q_cmd ##5##6##7
2146 {
2147 \str_if_eq:nnTF {##7} {##2}
2148 {
2149 \str_if_eq:onTF
2150 {##1} { } ##7 ##2 \q_cmd ##3 }
2151 { } {##2} { } }
2152 }
2153 { \use_i:nn }
2154 {
2155 ##1
2156 { __cmd_expandable_grab_D:NNNwNNnnn ##1##2##3##4 \q_cmd ##5##6 }
```

```

2157 \q_nil { } ##2 \ERROR \q__cmd \ERROR
2158 }
2159 {
2160 \msg_expandable_error:nnff { cmd } { missing-required }
2161 { \exp_args:Nf \tl_trim_spaces:n { \token_to_str:N ##5 } }
2162 { \tl_to_str:n {##2} }
2163 ##4 {#1} \q__cmd ##5 ##6 {##7}
2164 }
2165 }
2166 }
2167 \exp_args:No __cmd_tmp:w { \c_novalue_t1 }

(End of definition for __cmd_expandable_grab_R:w and __cmd_expandable_grab_R_aux:NNwNNn.)

```

When the delimiters are identical, nesting is not possible and a simplified approach is used. The test concept here is the same as for the case where the delimiters are different.

```

2168 \cs_new:Npn __cmd_expandable_grab_R_alt:w #1 \q__cmd #2#3
2169 { #2 { __cmd_expandable_grab_R_alt_aux:NNwNNn #1 \q__cmd #2#3 } }
2170 \cs_set_protected:Npn __cmd_tmp:w #1
2171 {
2172 \cs_new:Npn __cmd_expandable_grab_R_alt_aux:NNwNNn ##1##2##3 \q__cmd ##4##5##6
2173 {
2174 \str_if_eq:nnTF {##6} {##2}
2175 {
2176 \str_if_eq:onTF
2177 { ##1 { } ##6 ##2 ##2 }
2178 { { } ##2 }
2179 }
2180 { \use_i:i:nn }
2181 {
2182 ##1
2183 { __cmd_expandable_grab_D_alt:NNwn ##4 ##5 ##3 \q__cmd }
2184 ##6 \ERROR
2185 }
2186 {
2187 \msg_expandable_error:nnff { cmd } { missing-required }
2188 { \exp_args:Nf \tl_trim_spaces:n { \token_to_str:N ##4 } }
2189 { \tl_to_str:n {##2} }
2190 ##3 {#1} \q__cmd ##4 ##5 {##6}
2191 }
2192 }
2193 }
2194 \exp_args:No __cmd_tmp:w { \c_novalue_t1 }

(End of definition for __cmd_expandable_grab_R_alt:w and
__cmd_expandable_grab_R_alt_aux:NNwNNn.)

```

As for a D-type argument, here we compare the grabbed tokens using the only parser we have in order to work out if #2 is exactly equal to the output of the grabber.

```

2195 \cs_new:Npn __cmd_expandable_grab_t:w #1 \q__cmd #2#3
2196 { #2 { __cmd_expandable_grab_t_aux:NNwn #1 \q__cmd #2 #3 } }
2197 \cs_new:Npn __cmd_expandable_grab_t_aux:NNwn #1#2#3 \q__cmd #4#5#6
2198 {
2199 \str_if_eq:onTF { #1 { } #6 #2 } {#2}

```

```

2200 { #3 { \BooleanTrue } \q__cmd #4 #5 }
2201 { #3 { \BooleanFalse } \q__cmd #4 #5 {#6} }
2202 }

```

*(End of definition for `\_cmd_expandable_grab_t:w` and `\_cmd_expandable_grab_t_aux:NNwn`.)*

`\_cmd_put_arg_expandable:nw`  
`\_cmd_put_arg_expandable:ow` A useful helper, to store arguments when they are ready.

```

2203 \cs_new:Npn _cmd_put_arg_expandable:nw #1#2 \q__cmd { #2 {#1} \q__cmd }
2204 \cs_generate_variant:Nn _cmd_put_arg_expandable:nw { o }

```

*(End of definition for `\_cmd_put_arg_expandable:nw`.)*

## 1.10 Argument processors

`\_cmd_bool_reverse:N` A simple reversal.

```

2205 \cs_new_protected:Npn _cmd_bool_reverse:N #1
2206 {
2207 \bool_if:NTF #1
2208 { \tl_set:Nn \ProcessedArgument { \c_false_bool } }
2209 { \tl_set:Nn \ProcessedArgument { \c_true_bool } }
2210 }

```

*(End of definition for `\_cmd_bool_reverse:N`.)*

---

```
\l__cmd_split_list_seq
```

```
\l__cmd_split_list_tl
```

Splitting can take place either at a single token or at a longer identifier. To deal with single active tokens, a two-part procedure is needed.

```
2211 \seq_new:N \l__cmd_split_list_seq
2212 \tl_new:N \l__cmd_split_list_tl
2213 \cs_new_protected:Npn __cmd_split_list:nn #1#2
2214 {
2215 \tl_if_single:nTF {#1}
2216 {
2217 \token_if_cs:NTF #1
2218 { __cmd_split_list_multi:nn {#1} {#2} }
2219 { __cmd_split_list_single:Nn #1 {#2} }
2220 }
2221 { __cmd_split_list_multi:nn {#1} {#2} }
2222 }
2223 \cs_new_protected:Npn __cmd_split_list_multi:nn #1#2
2224 {
2225 \seq_set_split:Nnn \l__cmd_split_list_seq {#1} {#2}
2226 \tl_clear:N \ProcessedArgument
2227 \seq_map_inline:Nn \l__cmd_split_list_seq
2228 { \tl_put_right:Nn \ProcessedArgument { {##1} } }
2229 }
2230 \cs_generate_variant:Nn __cmd_split_list_multi:nn { nV }
2231 \group_begin:
2232 \char_set_catcode_active:N \^^@
2233 \cs_new_protected:Npn __cmd_split_list_single:Nn #1#2
2234 {
2235 \tl_set:Nn \l__cmd_split_list_tl {#2}
2236 \group_begin:
2237 \char_set_lccode:nn { '\^^@ } { '#1 }
2238 \tex_lowercase:D
2239 {
2240 \group_end:
2241 \tl_replace_all:Nnn \l__cmd_split_list_tl { ^@ }
2242 } {#1}
2243 __cmd_split_list_multi:nV {#1} \l__cmd_split_list_tl
2244 }
2245 \group_end:
```

(End of definition for `\__cmd_split_list:nn`, `\__cmd_split_list_multi:nn`, and  
`\__cmd_split_list_single:Nn`.)

```
__cmd_split_argument:nnn
```

```
__cmd_split_argument_aux:nnnn
```

```
__cmd_split_argument_aux:n
```

```
__cmd_split_argument_aux:wn
```

Splitting to a known number of items is a special version of splitting a list, in which the limit is hard-coded and where there will always be exactly the correct number of output items. An auxiliary function is used to save on working out the token list length several times.

```
2246 \cs_new_protected:Npn __cmd_split_argument:nnn #1#2#3
2247 {
2248 __cmd_split_list:nn {#2} {#3}
2249 \exp_args:Nf __cmd_split_argument_aux:nnnn
2250 { \tl_count:N \ProcessedArgument }
2251 {#1} {#2} {#3}
2252 }
```

```

2253 \cs_new_protected:Npn __cmd_split_argument_aux:nnnn #1#2#3#4
2254 {
2255 \int_compare:nNnF {#1} = { #2 + 1 }
2256 {
2257 \int_compare:nNnTF {#1} > { #2 + 1 }
2258 {
2259 \tl_set:Nx \ProcessedArgument
2260 {
2261 \exp_last_unbraced:NnNo
2262 __cmd_split_argument_aux:n
2263 { #2 + 1 }
2264 \use_none_delimit_by_q_stop:w
2265 \ProcessedArgument
2266 \q_stop
2267 }
2268 \msg_error:nxxxx { cmd } { arg-split }
2269 { \tl_to_str:n {#3} } { \int_eval:n { #2 + 1 } }
2270 { \tl_to_str:n {#4} }
2271 }
2272 {
2273 \tl_put_right:Nx \ProcessedArgument
2274 {
2275 \prg_replicate:nn { #2 + 1 - (#1) }
2276 { { \exp_not:V \c_novalue_tl } }
2277 }
2278 }
2279 }
2280 }
```

Auxiliaries to leave exactly the correct number of arguments in \ProcessedArgument.

```

2281 \cs_new:Npn __cmd_split_argument_aux:n #1
2282 { \prg_replicate:nn {#1} { __cmd_split_argument_aux:wn } }
2283 \cs_new:Npn __cmd_split_argument_aux:wn #1 \use_none_delimit_by_q_stop:w #2
2284 {
2285 \exp_not:n { {#2} }
2286 #1
2287 \use_none_delimit_by_q_stop:w
2288 }
```

(End of definition for \\_\_cmd\_split\_argument:nnn and others.)

\\_\_cmd\_trim\_spaces:n This one is almost trivial.

```

2289 \cs_new_protected:Npn __cmd_trim_spaces:n #1
2290 { \tl_set:Nx \ProcessedArgument { \tl_trim_spaces:n {#1} } }
```

(End of definition for \\_\_cmd\_trim\_spaces:n.)

## 1.11 Conversion to key–value form

This is implemented as a process but with no public interfaces, hence is treated separately from the others: it's a feature of \tcmd which just happens to use the same mechanism as a processor.

The two clear-cut cases have been eliminated, and we therefore have to deal with a search for = signs. We need an “action” loop here so we do not get misled by for example {=}. As the code here is for very much predictable types of input, we hard-code what constitutes math mode opening and closing. At the very beginning, the default key (#1) and the argument as given by the user (#2) are placed right after the \q\_\_cmd\_recursion\_stop, so that when the recursion ends, the macros \\_\_cmd\_arg\_to\_keyvalue\_set\_default:nn or \\_\_cmd\_arg\_to\_keyvalue\_set\_keyvalue:nn can be used to grab these two items and

set the \ProcessedArgument accordingly.

```

2332 \cs_new_protected:Npn __cmd_arg_to_keyvalue_auxv:nn #1#2
2333 {
2334 __cmd_arg_to_keyvalue_loop:w #2
2335 \q__cmd_recursion_tail \q__cmd_recursion_stop {#1} {#2}
2336 }
2337 \cs_new_protected:Npn __cmd_arg_to_keyvalue_loop:w #1 \q__cmd_recursion_stop
2338 {
2339 \tl_if_head_is_N_type:nTF {#1}
2340 { __cmd_arg_to_keyvalue_loop_N_type:N }
2341 {
2342 \tl_if_head_is_group:nTF {#1}
2343 { __cmd_arg_to_keyvalue_loop_group:n }
2344 { __cmd_arg_to_keyvalue_loop_space:w }
2345 }
2346 #1 \q__cmd_recursion_stop
2347 }
2348 \cs_new_protected:Npn __cmd_arg_to_keyvalue_loop_group:n #1
2349 { __cmd_arg_to_keyvalue_loop:w }
2350 \use:n { \cs_new_protected:Npn __cmd_arg_to_keyvalue_loop_space:w } ~
2351 { __cmd_arg_to_keyvalue_loop:w }
2352 \cs_new_protected:Npn __cmd_arg_to_keyvalue_loop_N_type:N #1
2353 {
2354 __cmd_if_recursion_tail_stop_do:Nn #1
2355 { __cmd_arg_to_keyvalue_set_default:nn }
2356 \str_if_eq:nnTF {#1} { = }
2357 {
2358 __cmd_use_i_delimit_by_q_recursion_stop:nw
2359 { __cmd_arg_to_keyvalue_set_keyvalue:nn }
2360 }
2361 {
2362 \bool_lazy_or:nnTF
2363 { \token_if_math_toggle_p:N #1 }
2364 { \str_if_eq_p:nn {#1} { \{ \} } }
2365 { __cmd_arg_to_keyvalue_math:w }
2366 { __cmd_arg_to_keyvalue_loop:w }
2367 }
2368 }
2369 \cs_new_protected:Npn __cmd_arg_to_keyvalue_math:w #1 \q__cmd_recursion_stop
2370 {
2371 \tl_if_head_is_N_type:nTF {#1}
2372 { __cmd_arg_to_keyvalue_math_N_type:N }
2373 {
2374 \tl_if_head_is_group:nTF {#1}
2375 { __cmd_arg_to_keyvalue_math_group:n }
2376 { __cmd_arg_to_keyvalue_math_space:w }
2377 }
2378 #1 \q__cmd_recursion_stop
2379 }
2380 \cs_new_protected:Npn __cmd_arg_to_keyvalue_math_N_type:N #1
2381 {
2382 __cmd_if_recursion_tail_stop_do:Nn #1
2383 { __cmd_arg_to_keyvalue_set_default:nn }
2384 \bool_lazy_or:nnTF

```

```

2385 { \token_if_math_toggle_p:N #1 }
2386 { \str_if_eq_p:nn {#1} { \) } }
2387 { __cmd_arg_to_keyvalue_loop:w }
2388 { __cmd_arg_to_keyvalue_math:w }
2389 }
2390 \cs_new_protected:Npn __cmd_arg_to_keyvalue_math_group:n #1
2391 { __cmd_arg_to_keyvalue_math:w }
2392 \use:n { \cs_new_protected:Npn __cmd_arg_to_keyvalue_math_space:w } ~
2393 { __cmd_arg_to_keyvalue_math:w }
2394 \cs_new_protected:Npn __cmd_arg_to_keyvalue_set_default:nn #1#2
2395 { \tl_set:Nn \ProcessedArgument { #1 = {#2} } }
2396 \cs_new_protected:Npn __cmd_arg_to_keyvalue_set_keyvalue:nn #1#2
2397 { \tl_set:Nn \ProcessedArgument {#2} }

```

A utility to allow us to grab the first N-type token without risking brace stripping the rest of the input.

```

2398 \cs_new:Npn __cmd_split_N_head_apply:Nn #1#2
2399 { \exp:w \if_false: { \fi: __cmd_split_N_head_apply_aux:NNw #1#2 } }
2400 \cs_new:Npn __cmd_split_N_head_apply_aux:NNw #1#2
2401 {
2402 \exp_after:wN \exp_end:
2403 \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN { \if_false: } \fi:
2404 }
2405

```

(End of definition for `\__cmd_arg_to_keyvalue:nn` and others.)

## 1.12 Utilities

`\__cmd_check_definable:nNT`  
`\__cmd_check_definable_aux:nN`

Check that a token list is appropriate as a first argument of `\NewDocumentCommand` and similar functions and otherwise produce an error. First trim whitespace to allow for spaces around the actual command to be defined. If the result has multiple tokens, it is not a valid argument. The single token is a control sequence exactly if its string representation has more than one character (using `\token_to_str:N` rather than `\tl_to_str:n` to avoid problems with macro parameter characters, and setting `\tex_escapechar:D` to prevent it from being non-printable). Finally, check for an active character: this is done by lowercasing the token to fix its character code (arbitrarily to that of `?`) and comparing the result to an active `?`. Both control sequences and active characters are valid arguments, and non-active character tokens are not. In all cases, the group opened to keep assignments local must be closed.

```

2406 \cs_new_protected:Npn __cmd_check_definable:nNT #1
2407 { \tl_trim_spaces_apply:nN {#1} __cmd_check_definable_aux:nN }
2408 \group_begin:
2409 \char_set_catcode_active:n { ‘? ’ }
2410 \cs_new_protected:Npn __cmd_check_definable_aux:nN #1#2
2411 {
2412 \group_begin:
2413 \tl_if_single_token:nTF {#1}
2414 {
2415 \int_set:Nn \tex_escapechar:D { 92 }
2416 \exp_args:Nx \tl_if_empty:nTF
2417 { \exp_args:No \str_tail:n { \token_to_str:N #1 } }
2418 }

```

```

2419 \exp_args:Nx \char_set_lccode:nn
2420 { ` \str_head:n {#1} } { '?' }
2421 \tex_lowercase:D { \tl_if_eq:nnTF {#1} } { ? }
2422 { \group_end: \use_iii:nnn }
2423 { \group_end: \use_i:nnn }
2424 }
2425 { \group_end: \use_iii:nnn }
2426 }
2427 { \group_end: \use_ii:nnn }
2428 {
2429 \msg_error:nnxx { cmd } { not-definable }
2430 { \tl_to_str:n {#1} } { \token_to_str:N #2 }
2431 }
2432 {
2433 \msg_error:nnxx { cmd } { not-one-token }
2434 { \tl_to_str:n {#1} } { \token_to_str:N #2 }
2435 }
2436 }
2437 \group_end:

```

(End of definition for `\__cmd_check_definable:nNT` and `\__cmd_check_definable_aux:nN`.)

`\__cmd_token_if_cs:NTF`

Based on the definition of `\__cmd_check_definable_aux:nN` above, but only checks for an actual control sequence (*i.e.*, `\anything`). `\tex_escapechar:D` is temporarily changed to a known value and then it checks if `\string#1` contains more than one character: if it does, it's a control sequence. This test differs from `\token_if_cs:NTF` for example in `\token_if_cs:NTF \c_group_begin_token {T}{F}`, where `\token_if_cs:NTF` returns false.

```

2438 \cs_new_protected:Npn __cmd_token_if_cs:NTF #1
2439 {
2440 \group_begin:
2441 \int_set:Nn \tex_escapechar:D { 92 }
2442 \exp_args:Nx \tl_if_empty:nTF
2443 { \exp_args:No \str_tail:n { \token_to_str:N #1 } }
2444 { \group_end: \use_ii:nn }
2445 { \group_end: \use_i:nn }
2446 }

```

(End of definition for `\__cmd_token_if_cs:NTF`.)

Analogue of `\seq_mapthread_function:NNN` for token lists.

```

2447 \cs_new:Npn __cmd_tl_mapthread_function:NNN #1#2#3
2448 {
2449 \exp_after:wN \exp_after:wN
2450 \exp_after:wN __cmd_tl_mapthread_loop:w
2451 \exp_after:wN \exp_after:wN
2452 \exp_after:wN #3
2453 \exp_after:wN #1
2454 \exp_after:wN \q_recursion_tail
2455 \exp_after:wN \q_mark
2456 #2
2457 \q_recursion_tail
2458 \q_recursion_stop
2459 }

```

```

2460 \cs_new:Npn __cmd_tl_mapthread_function:nnN #1#2#3
2461 {
2462 __cmd_tl_mapthread_loop:w #3
2463 #1 \q_recursion_tail \q_mark
2464 #2 \q_recursion_tail \q_recursion_stop
2465 }
2466 \cs_new:Npn __cmd_tl_mapthread_loop:w #1#2#3 \q_mark #4
2467 {
2468 \quark_if_recursion_tail_stop:n {#2}
2469 \quark_if_recursion_tail_stop:n {#4}
2470 #1 {#2} {#4}
2471 __cmd_tl_mapthread_loop:w #1#3 \q_mark
2472 }

```

(End of definition for `\__cmd_tl_mapthread_function:NNN`, `\__cmd_tl_mapthread_function:nnN`, and `\__cmd_tl_mapthread_loop:w`.)

`\__kernel_cmd_if_xparse:NTF`  
`\__cmd_cmd_type_cases:NnnnnF`  
`\__cmd_cmd_if_xparse_aux:N`

To determine whether the command is an `xparse` command check that its `arg_spec` is empty (this also excludes non-macros) and that its `replacement_spec` starts with either `\__cmd_start:nNNnnn` (non-expandable command) or `\__cmd_start_expandable:nNNNNn` (expandable command) or `\__cmd_start_optimized:` (optimized command) or `\__cmd_start_env:nnnnn` (environment) or `\environment #1 end aux` (environment end).

This conditional is needed in several kernel modules and is therefore has a kernel-internal name.

```

2473 \cs_new_protected:Npn __cmd_cmd_type_cases:NnnnnnF #1 #2 #3 #4 #5 #6 #7
2474 {
2475 \exp_args:Ne \str_case_e:nnF
2476 {
2477 \exp_args:Nf \tl_if_empty:nT { __kernel_cs_parameter_spec:N #1 }
2478 { \exp_not:N \exp_not:n { \exp_not:e { \tl_head:N #1 } } }
2479 }
2480 {
2481 { \exp_not:N __cmd_start:nNNnnn } {#2}
2482 { \exp_not:N __cmd_start_expandable:nNNNNn } {#3}
2483 { \exp_not:N __cmd_start_optimized: } {#4}
2484 { \exp_not:N __cmd_start_env:nnnnn } {#5}
2485 {
2486 \exp_after:wN \exp_not:N
2487 \cs:w environment-
2488 \exp_last_unbraced:Ne \use_none:nnn
2489 { \cs_to_str:N #1 } ~end-aux \cs_end:
2490 } {#6}
2491 }
2492 {#7}
2493 }
2494 \cs_new_protected:Npn __kernel_cmd_if_xparse:NTF #1
2495 {
2496 __cmd_cmd_type_cases:NnnnnnF #1
2497 { } { } { } { } { } { } { \use_iii:nnn }
2498 \use_i:nn
2499 }

```

(End of definition for `\__kernel_cmd_if_xparse:NTF`, `\__cmd_cmd_type_cases:NnnnnnF`, and `\__cmd_cmd_if_xparse_aux:N`.)

\\_\\_cmd\\_peek\\_nonspace:NTF  
\\_\\_cmd\\_peek\\_nonspace\\_remove:NTF  
\\_\\_cmd\\_peek\\_nonspace\\_aux:nNNTF

Collect spaces in a loop, and put the collected spaces back in the false branch of a call to \peek\\_meaning:NTF or \peek\\_meaning\\_remove:NTF.

```

2500 \cs_new_protected:Npn __cmd_peek_nonspace:NTF
2501 { __cmd_peek_nonspace_aux:nNNTF { } __cmd_peek_meaning:NTF }
2502 \cs_new_protected:Npn __cmd_peek_nonspace_remove:NTF
2503 { __cmd_peek_nonspace_aux:nNNTF { } __cmd_peek_meaning_remove:NTF }
2504 \cs_new_protected:Npn __cmd_peek_nonspace_aux:nNNTF #1#2#3#4#5
2505 {
2506 \peek_meaning_remove:NTF \c_space_token
2507 { __cmd_peek_nonspace_aux:nNNTF { #1 ~ } #2 #3 {#4} {#5} }
2508 { #2 #3 { #4 } { #5 #1 } }
2509 }

```

(End of definition for \\_\\_cmd\\_peek\\_nonspace:NTF, \\_\\_cmd\\_peek\\_nonspace\\_remove:NTF, and \\_\\_cmd\\_peek\\_nonspace\\_aux:nNNTF.)

\\_\\_cmd\\_peek\\_meaning:NTF  
\\_\\_cmd\\_peek\\_meaning\\_remove:NTF  
\\_\\_cmd\\_peek\\_cs\\_check\\_equal:NNN  
\\_\\_cmd\\_peek\\_meaning\\_aux:NNTF  
\\_\\_cmd\\_peek\\_true\\_remove:Nw

Peek ahead for a token with a given meaning. In case the search token is a control sequence, also check that the <csname> is the same as the control sequence peeked at. This extra verification is necessary when the command is delimited by control sequence tokens (as opposed to character tokens), and we want the exact same control sequence to match.

```

2510 \cs_new_protected:Npn __cmd_peek_meaning:NTF
2511 { __cmd_peek_meaning_aux:NNTF \c_false_bool }
2512 \cs_new_protected:Npn __cmd_peek_meaning_remove:NTF
2513 { __cmd_peek_meaning_aux:NNTF \c_true_bool }
2514 \cs_new_protected:Npn __cmd_peek_meaning_aux:NNTF #1#2#3#4
2515 {
2516 \tl_set:Nn \l__cmd_tma_t1 {#3}
2517 \tl_set:Nn \l__cmd_tmrb_t1 {#4}
2518 \peek_meaning:NTF #2
2519 {
2520 \token_if_eq_meaning:NNTF #2 \c_group_begin_token
2521 { __cmd_peek_true_remove:Nw #1 }
2522 {
2523 __cmd_token_if_cs:NTF #2
2524 { __cmd_peek_cs_check_equal:NNN #1 #2 }
2525 { __cmd_peek_true_remove:Nw #1 }
2526 }
2527 }
2528 { \l__cmd_tmrb_t1 }
2529 }
2530 \cs_new_protected:Npn __cmd_peek_cs_check_equal:NNN #1#2#3
2531 {
2532 \str_if_eq:nnTF {#2} {#3}
2533 { __cmd_peek_true_remove:Nw #1 }
2534 { \l__cmd_tmrb_t1 }
2535 #3
2536 }
2537 \cs_new_protected:Npn __cmd_peek_true_remove:Nw #1
2538 {
2539 \bool_if:NTF #1
2540 {
2541 \tex_afterassignment:D \l__cmd_tma_t1
2542 \cs_set_eq:NN __cmd_tmp:w

```

```

2543 }
2544 { \l__cmd_tmpa_tl }
2545 }

```

(End of definition for `\_cmd_peek_meaning:NTF` and others.)

## 1.13 Messages

---

```

\c__cmd_ignore_def_tl 2546 \tl_const:Nn \c__cmd_ignore_def_tl
2547 { \\ \\ LaTeX-will-ignore-this-entire-definition. }

```

`\_cmd_environment_or_command`: Two texts used in several messages.

```

2548 \cs_new:Npn _cmd_environment_or_command:
2549 {
2550 \bool_if:NTF \l__cmd_environment_bool
2551 { environment ~' \l__cmd_environment_str ' }
2552 {
2553 command ~
2554 ' \c_backslash_str \tl_to_str:N \l__cmd_function_tl '
2555 }
2556 }

```

(End of definition for `\_cmd_environment_or_command`.)

Some messages intended as errors when defining commands/environments.

```

2557 \msg_new:nnnn { cmd } { arg-after-body }
2558 { Argument-type-'b'-must-be-last-in-#1. }
2559 {
2560 The-'b'-argument-type-must-come-last-but-it-is-followed-
2561 by-'#2'-in-the-argument-specification.-This-is-not-allowed.
2562 \c__cmd_ignore_def_tl
2563 }
2564 \msg_new:nnnn { cmd } { bad-arg-spec }
2565 { Bad-argument-specification-'#2'-for-#1. }
2566 {
2567 The-argument-specification-provided-is-not-valid:-
2568 one-or-more-mandatory-parts-are-missing.
2569 \c__cmd_ignore_def_tl
2570 }
2571 \msg_new:nnnn { cmd } { already-defined }
2572 { Command-'#1'-already-defined. }
2573 {
2574 You-have-used-#2-
2575 with-a-command-that-already-has-a-definition. \\ \\
2576 The-existing-definition-of-'#1'-will-not-be-altered.
2577 }
2578 \msg_new:nnnn { cmd } { undefined }
2579 { Command ~'#1'~undefined. }
2580 {
2581 You-have-used-#2-
2582 with-a-command-that-was-never-defined.
2583 \c__cmd_ignore_def_tl

```

```

2584 }
2585 \msg_new:nnnn { cmd } { env-already-defined }
2586 { Environment~'#1'~already~defined. }
2587 {
2588 You~have~used~\NewDocumentEnvironment
2589 with~an~environment~that~already~has~a~definition. \\ \\
2590 The~existing~definition~of~'#1'~will~not~be~altered.
2591 }
2592 \msg_new:nnnn { cmd } { env-end-already-defined }
2593 { End~of~environment~'#1'~already~defined. }
2594 {
2595 You~have~used~\NewDocumentEnvironment
2596 with~an~environment~that~already~has~a~definition~for~'end#1'. \\ \\
2597 The~existing~definition~of~'#1'~will~not~be~altered.
2598 }
2599 \msg_new:nnnn { cmd } { env-undefined }
2600 { Environment~'#1'~undefined. }
2601 {
2602 You~have~used~\RenewDocumentEnvironment
2603 with~an~environment~that~was~never~defined.
2604 \c__cmd_ignore_def_tl
2605 }
2606 \msg_new:nnnn { cmd } { expandable-ending-optional }
2607 { Bad~argument~specification~'#2'~for~'#1'. }
2608 {
2609 Expandable~commands~must~have~a~final~mandatory~argument~
2610 (or~no~arguments~at~all).~You~cannot~have~a~terminal~optional~
2611 argument~with~expandable~commands.
2612 }
2613 \msg_new:nnnn { cmd } { long-short-mix }
2614 { Invalid~argument~prefix~'+'~in~command~'#1'. }
2615 {
2616 The~arguments~for~an~expandable~command~must~not~involve~short~
2617 arguments~after~long~arguments.~You~have~tried~to~mix~the~two~types~
2618 when~defining~'#1'.
2619 }
2620 \msg_new:nnnn { cmd } { invalid-command-arg }
2621 { Invalid~argument~type~'#2'~in~'#1'. }
2622 {
2623 The~letter~'#2'~can~only~be~used~in~environment~argument~
2624 specifications,~but~not~for~commands.
2625 \\ \\
2626 LaTeX~will~ignore~the~entire~definition.
2627 }
2628 \msg_new:nnnn { cmd } { invalid-expandable-arg }
2629 { Invalid~argument~type~'#2'~in~'#1'. }
2630 {
2631 The~letter~'#2'~specifies~an~argument~type~which~cannot~be~used~
2632 in~an~expandable~command.
2633 \c__cmd_ignore_def_tl
2634 }
2635 \msg_new:nnnn { cmd } { invalid-after-optional-expandably }
2636 { Argument~'#2'~invalid~after~optional~arg~in~'#1'. }
2637 {

```

```

2638 The~letter~'#2'~specifies~an~argument~type~which~cannot~be~used~
2639 in~an~expandable~command~after~an~optional~argument.
2640 \c__cmd_ignore_def_tl
2641 }
2642 \msg_new:nnnn { cmd } { invalid-bang }
2643 { Invalid~argument~prefix~'!'~in~#1. }
2644 {
2645 The~prefix~'!'~is~only~allowed~for~trailing~optional~arguments.~
2646 You~tried~to~apply~it~to~#2.
2647 \c__cmd_ignore_def_tl
2648 }
2649 \msg_new:nnnn { cmd } { not-definable }
2650 { First~argument~of~'#2'~must~be~a~command. }
2651 {
2652 The~first~argument~of~'#2'~should~be~the~document~command~that~will~
2653 be~defined.~The~provided~argument~'#1'~is~a~character.~Perhaps~a~
2654 backslash~is~missing?
2655 \c__cmd_ignore_def_tl
2656 }
2657 \msg_new:nnnn { cmd } { not-one-token }
2658 { First~argument~of~'#2'~must~be~a~command. }
2659 {
2660 The~first~argument~of~'#2'~should~be~the~document~command~that~will~
2661 be~defined.~The~provided~argument~'#1'~contains~more~than~one~
2662 token.~Perhaps~a~backslash~is~missing?
2663 \c__cmd_ignore_def_tl
2664 }
2665 \msg_new:nnnn { cmd } { not-single-token }
2666 { Argument~delimiter~'#2'~invalid~in~#1. }
2667 {
2668 The~argument~specification~contains~
2669 \tl_if_empty:nTF{#2}{nothing}{'#2'}~
2670 in~a~place~
2671 where~a~single~token~is~required.
2672 \c__cmd_ignore_def_tl
2673 }
2674 \msg_new:nnnn { cmd } { forbidden-group-token }
2675 { Argument~delimiter~'#2'~invalid~in~#1. }
2676 {
2677 The~argument~specification~contains~the~implicit~
2678 #3-group~token~'#2'~which~is~not~allowed~as~an~argument~delimiter.
2679 \c__cmd_ignore_def_tl
2680 }
2681 \msg_new:nnnn { cmd } { processor-in-expandable }
2682 { Invalid~argument~prefix~'>'~in~command~'#1'. }
2683 {
2684 The~argument~specification~for~'#1'~contains~the~processor~function~'>{#2}'.~
2685 This~is~only~supported~for~robust~commands,~but~not~for~expandable~ones.
2686 \c__cmd_ignore_def_tl
2687 }
2688 \msg_new:nnnn { cmd } { keyval-in-expandable }
2689 { Invalid~argument~prefix~'='~in~command~'#1'. }
2690 {
2691 The~argument~specification~for~'#1'~contains~a~key--value~marker~'={#2}'.~

```

```

2692 This~is~only~supported~for~robust~commands,~but~not~for~expandable~ones.
2693 \c__cmd_ignore_def_tl
2694 }
2695 \msg_new:nnnn { cmd } { too-many-args }
2696 { Too~many~arguments~for~#1. }
2697 {
2698 The~argument~specification~'#2'~asks~for~more~than~9~arguments.~
2699 This~cannot~be~implemented.
2700 \c__cmd_ignore_def_tl
2701 }
2702 \msg_new:nnnn { cmd } { two-markers }
2703 { Invalid~argument~prefix~'#2'~in~#1. }
2704 {
2705 The~argument~specification~provided~for~#1~has~two~'#2'~markers~applied~
2706 to~the~same~argument;~one~is~redundant.
2707 }
2708 \msg_new:nnnn { cmd } { unknown-argument-type } % should be unkown-arg-type but dep in xparses
2709 { Invalid~argument~type~'#2'~in~#1. }
2710 {
2711 The~letter~'#2'~does~not~specify~a~known~argument~type.
2712 \c__cmd_ignore_def_tl
2713 }
2714 \msg_new:nnnn { cmd } { xparses-arg-type }
2715 { Invalid~argument~type~'#2'~in~#1~(requires~xparses). }
2716 {
2717 The~letter~'#2'~specifies~a~known~but~deprecated~argument~type.~
2718 If~you~really~need~it~you~have~to~load~the~xparses~package.
2719 \c__cmd_ignore_def_tl
2720 }

```

Errors when using commands/environments. The `if-boolean` message is always used in expandable errors. The `default-loop` and `missing-required` messages can be expandable or not expandable.

```

2721 \msg_new:nnn { cmd } { if-boolean }
2722 { Invalid~argument~{#1}~to~\iow_char:N\\IfBoolean... }
2723 \msg_new:nnnn { cmd } { default-loop }
2724 { Circular~dependency~in~defaults~of~#1. }
2725 {
2726 The~default~values~of~two~or~more~arguments~of~the~#1~
2727 depend~on~each~other~in~a~way~that~cannot~be~resolved.
2728 }
2729 \msg_new:nnnn { cmd } { missing-required }
2730 { Required~argument~missing~for~#1. }
2731 {
2732 The~#1~expects~one~of~its~arguments~to~start~with~'#2'.~
2733 LaTeX~did~not~find~this~argument~and~will~insert~a~default~value~
2734 for~further~processing.
2735 }
2736 \msg_new:nnnn { cmd } { arg-split }
2737 { Too~many~'#1'~separators~in~argument. }
2738 {
2739 LaTeX~was~asked~to~split~the~input~'#3'~
2740 at~each~occurrence~of~the~separator~'#1'~into~#2-parts.~
2741 Too~many~separators~were~found.

```

```

2742 }
2743 \msg_new:n { cmd } { verbatim-nl }
2744 { Verbatim-like~#1-ended~by~end~of~line. }
2745 {
2746 The~verbatim~argument~of~the~#1~cannot~contain~more~than~one~line,~
2747 but~the~end~

2748 of~the~current~line~has~been~reached.~You~may~have~forgotten~the~

2749 closing~delimiter.

2750 \\ \\
2751 LaTeX~will~ignore~'#2'~and~you~may~get~some~additional~

2752 (low-level)~errors.
2753 }
2754 \msg_new:n { cmd } { verbatim-tokenized }
2755 { Verbatim-like~#1-illegal~in~argument. }
2756 {
2757 The~#1-takes~a~verbatim~argument~and~should~therefore~normally~

2758 not~be~used~in~arguments~of~other~commands~or~environments.~

2759 LaTeX~found~an~illegal~token~ \tl_if_empty:nF {#3} { (#3)~ }

2760 after~'#2'~and~will~drop~everything~up~to~this~point.

2761 \\ \\
2762 Expect~further~(low-level)~errors.
2763 }

Intended more for information.

2764 \msg_new:n { cmd } { define-command } % should be just ‘‘define’’ but dep in xparse
2765 {
2766 Defining~command~#1~

2767 with~sig.~'#2'~\msg_line_context:.
2768 }
2769 \msg_new:n { cmd } { define-env }
2770 {
2771 Defining~environment~#1~

2772 with~sig.~'#2'~\msg_line_context:.
2773 }
2774 \msg_new:n { cmd } { redefine }
2775 {
2776 Redefining~command~#1~

2777 with~sig.~'#2'~\msg_line_context:.
2778 }
2779 \msg_new:n { cmd } { redefine-env }
2780 {
2781 Redefining~environment~#1~

2782 with~sig.~'#2'~\msg_line_context:.
2783 }
2784 \msg_new:n { cmd } { optional-mandatory }
2785 {
2786 Optional~and~mandatory~argument~with~same~delimiter~'#2'.
2787 \\ \\
2788 The~mandatory~argument~specified~with~

2789 '\str_case:nnF{#1}{ {R/r}{r'~or~'R} }{#1}'~has~the~

2790 same~delimiter~'#2'~as~an~earlier~optional~argument.~

2791 It~will~therefore~not~be~possible~to~omit~all~the~earlier~

2792 optional~arguments~when~calling~this~command.

2793 \\ \\
2794 This~may~be~intentional,~but~then~it~might~be~a~mistake.

```

```

2795 }
2796 \msg_new:n { cmd } { unsupported-let }
2797 {
2798 The~command~'#1'~was~undefined~but~not~the~associated~commands~
2799 '#1~code'~and/or~'#1~defaults'.~Maybe~you~tried~using~
2800 \iow_char:N\let.~This~may~lead~to~an~infinite~loop.
2801 }
```

## 1.14 User functions

The user functions are more or less just the internal functions renamed.

\BooleanFalse Design-space names for the Boolean values.

```

2802 \cs_new_eq:NN \BooleanFalse \c_false_bool
2803 \cs_new_eq:NN \BooleanTrue \c_true_bool
```

*(End of definition for \BooleanFalse and \BooleanTrue.)*

\NewDocumentCommand \RenewDocumentCommand \ProvideDocumentCommand The user macros are pretty simple wrappers around the internal ones. There is however a check that the first argument is a single token, possibly surrounded by spaces (hence the strange \use:n), and is definable.

```

2804 \cs_new_protected:Npn \NewDocumentCommand #1#2#3
2805 {
2806 __cmd_check_definable:nNT {#1} \NewDocumentCommand
2807 {
2808 \cs_if_exist:NTF #1
2809 {
2810 \msg_error:nnxx { cmd } { already-defined }
2811 { \use:n { \token_to_str:N #1 { } } }
2812 { \token_to_str:N \NewDocumentCommand }
2813 }
2814 { __cmd_declare_cmd:Nnn #1 {#2} {#3} }
2815 }
2816 }
2817 \cs_new_protected:Npn \RenewDocumentCommand #1#2#3
2818 {
2819 __cmd_check_definable:nNT {#1} \RenewDocumentCommand
2820 {
2821 \cs_if_exist:NTF #1
2822 { __cmd_declare_cmd:Nnn #1 {#2} {#3} }
2823 {
2824 \msg_error:nnxx { cmd } { undefined }
2825 { \use:n { \token_to_str:N #1 { } } }
2826 { \token_to_str:N \RenewDocumentCommand }
2827 }
2828 }
2829 }
2830 \cs_new_protected:Npn \ProvideDocumentCommand #1#2#3
2831 {
2832 __cmd_check_definable:nNT {#1} \ProvideDocumentCommand
2833 { \cs_if_exist:NF #1 { __cmd_declare_cmd:Nnn #1 {#2} {#3} } }
2834 }
2835 \cs_new_protected:Npn \DeclareDocumentCommand #1#2#3
2836 {
```

```

2837 __cmd_check_definable:nNT {#1} \DeclareDocumentCommand
2838 { __cmd_declare_cmd:Nnn #1 {#2} {#3} }
2839 }

(End of definition for \NewDocumentCommand and others.)

\NewDocumentEnvironment
\RenewDocumentEnvironment
\ProvideDocumentEnvironment
\DeclareDocumentEnvironment
 __cmd_new_env:nnnn
 __cmd_renew_env:nnnn
 __cmd_provide_env:nnnn
 __cmd_new_env:ennn
 __cmd_renew_env:ennn
 __cmd_provide_env:ennn

Very similar for environments. Trim spaces from user-specified <envname>, do existence
check then hand off to __cmd_declare_env:nnnn.

2840 ⟨latexrelease⟩\IncludeInRelease{2024/11/01}{\NewDocumentEnvironment}%
2841 ⟨latexrelease⟩
2842 {\Trim~spaces~from~envname~first}
\cs_new_protected:Npn \NewDocumentEnvironment #1#2#3#4
{
 __cmd_new_env:ennn { \tl_trim_spaces:e {#1} } {#2} {#3} {#4}
}
\cs_new_protected:Npn \RenewDocumentEnvironment #1#2#3#4
{
 __cmd_renew_env:ennn { \tl_trim_spaces:e {#1} } {#2} {#3} {#4}
}
\cs_new_protected:Npn \ProvideDocumentEnvironment #1#2#3#4
{
 __cmd_provide_env:ennn { \tl_trim_spaces:e {#1} } {#2} {#3} {#4}
}
\cs_new_protected:Npn \DeclareDocumentEnvironment #1#2#3#4
{
 __cmd_declare_env:ennn { \tl_trim_spaces:e {#1} } {#2} {#3} {#4}
}

2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857

Each of __cmd_(new|renew|provide)_env:nnnn is curried.

2858 \cs_new_protected:Npn __cmd_new_env:nnnn #1
2859 {
2860 \cs_if_exist:cTF {#1}
2861 {
2862 \msg_error:nnx { cmd } { env-already-defined } {#1}
2863 \use_none:nnn
2864 }
2865 {
2866 \cs_if_exist:cTF { end #1 }
2867 {
2868 \msg_error:nnx { cmd } { env-end-already-defined } {#1}
2869 \use_none:nnn
2870 }
2871 { __cmd_declare_env:nnnn {#1} }
2872 }
2873 }
2874 \cs_new_protected:Npn __cmd_renew_env:nnnn #1
2875 {
2876 \cs_if_exist:cTF {#1}
2877 { __cmd_declare_env:nnnn {#1} }
2878 {
2879 \msg_error:nnx { cmd } { env-undefined } {#1}
2880 \use_none:nnn
2881 }
2882 }
2883 \cs_new_protected:Npn __cmd_provide_env:nnnn #1
2884 {

```

```

2885 \cs_if_exist:cTF {#1}
2886 { \use_none:nnn }
2887 { _cmd_declare_env:nnnn {#1} }
2888 }
2889 \cs_generate_variant:Nn _cmd_new_env:nnnn { e }
2890 \cs_generate_variant:Nn _cmd_renew_env:nnnn { e }
2891 \cs_generate_variant:Nn _cmd_provide_env:nnnn { e }
2892 \langle latexrelease \rangle \EndIncludeInRelease
2893 \langle latexrelease \rangle \IncludeInRelease{2024/06/01}{\NewDocumentEnvironment}%
2894 \langle latexrelease \rangle \Trim~spaces~from~envname~first
2895 \langle latexrelease \rangle \cs_new_protected:Npn \NewDocumentEnvironment #1#2#3#4
2896 \langle latexrelease \rangle {
2897 \langle latexrelease \rangle \cs_if_exist:cTF {#1}
2898 \langle latexrelease \rangle { \msg_error:nnx { cmd } { env-already-defined } {#1} }
2899 \langle latexrelease \rangle {
2900 \langle latexrelease \rangle \cs_if_exist:cTF { end #1 }
2901 \langle latexrelease \rangle { \msg_error:nnx { cmd } { env-end-already-defined } {#1} }
2902 \langle latexrelease \rangle { _cmd_declare_env:nnnn {#1} {#2} {#3} {#4} }
2903 \langle latexrelease \rangle }
2904 \langle latexrelease \rangle }
2905 \langle latexrelease \rangle \cs_new_protected:Npn \RenewDocumentEnvironment #1#2#3#4
2906 \langle latexrelease \rangle {
2907 \langle latexrelease \rangle \cs_if_exist:cTF {#1}
2908 \langle latexrelease \rangle { _cmd_declare_env:nnnn {#1} {#2} {#3} {#4} }
2909 \langle latexrelease \rangle { \msg_error:nnx { cmd } { env-undefined } {#1} }
2910 \langle latexrelease \rangle }
2911 \langle latexrelease \rangle \cs_new_protected:Npn \ProvideDocumentEnvironment #1#2#3#4
2912 \langle latexrelease \rangle { \cs_if_exist:cF {#1} { _cmd_declare_env:nnnn {#1} {#2} {#3} {#4} } }
2913 \langle latexrelease \rangle \cs_new_protected:Npn \DeclareDocumentEnvironment #1#2#3#4
2914 \langle latexrelease \rangle { _cmd_declare_env:nnnn {#1} {#2} {#3} {#4} }
2915 \langle latexrelease \rangle \cs_undefine:N _cmd_new_env:nnnn
2916 \langle latexrelease \rangle \cs_undefine:N _cmd_new_env:ennn
2917 \langle latexrelease \rangle \cs_undefine:N _cmd_renew_env:nnnn
2918 \langle latexrelease \rangle \cs_undefine:N _cmd_renew_env:ennn
2919 \langle latexrelease \rangle \cs_undefine:N _cmd_provide_env:nnnn
2920 \langle latexrelease \rangle \cs_undefine:N _cmd_provide_env:ennn
2921 \langle latexrelease \rangle
2922 \langle latexrelease \rangle \EndIncludeInRelease

```

(End of definition for `\NewDocumentEnvironment` and others.)

`\NewExpandableDocumentCommand`  
`\RenewExpandableDocumentCommand`  
`\ProvideExpandableDocumentCommand`  
`\DeclareExpandableDocumentCommand`

```

2923 \cs_new_protected:Npn \NewExpandableDocumentCommand #1#2#3
2924 {
2925 _cmd_check_definable:nNT {#1} \NewExpandableDocumentCommand
2926 {
2927 \cs_if_exist:NTF #1
2928 {
2929 \msg_error:nnxx { cmd } { already-defined }
2930 { \use:nnn \token_to_str:N #1 { } }
2931 { \token_to_str:N \NewExpandableDocumentCommand }
2932 }

```

```

2933 { __cmd_declare_expandable_cmd:Nnn #1 {#2} {#3} }
2934 }
2935 }
2936 \cs_new_protected:Npn \RenewExpandableDocumentCommand #1#2#3
2937 {
2938 __cmd_check_definable:nNT {#1} \RenewExpandableDocumentCommand
2939 {
2940 \cs_if_exist:NTF #1
2941 { __cmd_declare_expandable_cmd:Nnn #1 {#2} {#3} }
2942 {
2943 \msg_error:nnxx { cmd } { undefined }
2944 { \use:nnn \token_to_str:N #1 { } }
2945 { \token_to_str:N \RenewExpandableDocumentCommand }
2946 }
2947 }
2948 }
2949 \cs_new_protected:Npn \ProvideExpandableDocumentCommand #1#2#3
2950 {
2951 __cmd_check_definable:nNT {#1} \ProvideExpandableDocumentCommand
2952 {
2953 \cs_if_exist:NF #1
2954 { __cmd_declare_expandable_cmd:Nnn #1 {#2} {#3} }
2955 }
2956 }
2957 \cs_new_protected:Npn \DeclareExpandableDocumentCommand #1#2#3
2958 {
2959 __cmd_check_definable:nNT {#1} \DeclareExpandableDocumentCommand
2960 { __cmd_declare_expandable_cmd:Nnn #1 {#2} {#3} }
2961 }

```

(End of definition for `\NewExpandableDocumentCommand` and others.)

`\IfBooleanT` The logical `<true>` and `<false>` statements are just the normal `\c_true_bool` and `\c_false_bool` so `\bool_if:NTF` is almost enough. However, this code-level function blows up badly when passed invalid input. We want `\IfBooleanTF` to accept a single (non-space) token equal to `\c_true_bool` or `\c_false_bool`, possibly surrounded by spaces. If the input is blank or multiple items, jump to the error and pick the false branch. If the input, ignoring spaces (we do this by omitting braces in the `\tl_if_single_token:nF` test), is not a single token then jump to the error as well. It is then safe to compare the token to the two booleans, picking the appropriate branch. If neither matches, we jump to the error as well.

```

2962 \cs_new:Npn \IfBooleanTF #1
2963 {
2964 \tl_if_single:nF {#1}
2965 { \prg_break:n { \use:n } }
2966 \tl_if_single_token:nF #1
2967 { \prg_break:n { \use:n } }
2968 \token_if_eq_meaning:NNT #1 \c_true_bool
2969 { \prg_break:n { \use_ii:nnn } }
2970 \token_if_eq_meaning:NNT #1 \c_false_bool
2971 { \prg_break:n { \use_iii:nnn } }
2972 \prg_break:n { \use:n }
2973 \prg_break_point:
2974 }

```

```

2975 \msg_expandable_error:nnn { cmd } { if-boolean } {#1}
2976 \use_ii:nn
2977 }
2978 }
2979 \cs_new:Npn \IfBooleanT #1#2 { \IfBooleanTF {#1} {#2} { } }
2980 \cs_new:Npn \IfBooleanF #1 { \IfBooleanTF {#1} { } }
(End of definition for \IfBooleanT, \IfBooleanF, and \IfBooleanTF.)

\IfNoValueT Simple re-naming.
\IfNoValueF 2981 \cs_new_eq:NN \IfNoValueF \tl_if_novalue:nF
\IfNoValueTF 2982 \cs_new_eq:NN \IfNoValueT \tl_if_novalue:nT
2983 \cs_new_eq:NN \IfNoValueTF \tl_if_novalue:nTF
(End of definition for \IfNoValueT, \IfNoValueF, and \IfNoValueTF.)

\IfValueT Inverted logic.
\IfValueF 2984 \cs_new:Npn \IfValueF { \tl_if_novalue:nT }
\IfValueTF 2985 \cs_new:Npn \IfValueT { \tl_if_novalue:nF }
2986 \cs_new:Npn \IfValueTF #1#2#3 { \tl_if_novalue:nTF {#1} {#3} {#2} }
(End of definition for \IfValueT, \IfValueF, and \IfValueTF.)

\IfBlankT Another simple re-naming.
\IfBlankF 2987 ⟨latexrelease⟩\IncludeInRelease{2022/06/01}%
\IfBlankTF 2988 ⟨latexrelease⟩ {\IfBlankTF}{Testing~for~empty~or~blank}%
2989 \cs_new_eq:NN \IfBlankF \tl_if_blank:nF
2990 \cs_new_eq:NN \IfBlankT \tl_if_blank:nT
2991 \cs_new_eq:NN \IfBlankTF \tl_if_blank:nTF
2992 ⟨latexrelease⟩\EndIncludeInRelease
2993 ⟨latexrelease⟩\IncludeInRelease{2021/11/15}%
2994 ⟨latexrelease⟩ {\IfBlankTF}{Testing~for~empty~or~blank}%
2995 ⟨latexrelease⟩\cs_undefine:N \IfBlankF
2996 ⟨latexrelease⟩\cs_undefine:N \IfBlankT
2997 ⟨latexrelease⟩\cs_undefine:N \IfBlankTF
2998 ⟨latexrelease⟩
2999 ⟨latexrelease⟩\EndIncludeInRelease
(End of definition for \IfBlankT, \IfBlankF, and \IfBlankTF.)

\ProcessedArgument Processed arguments are returned using this name, which is reserved here although the
definition will change.
3000 \tl_new:N \ProcessedArgument
(End of definition for \ProcessedArgument.)

\ReverseBoolean Simple copies.
\SplitArgument 3001 \cs_new_eq:NN \ReverseBoolean __cmd_bool_reverse:N
\SplitList 3002 \cs_new_eq:NN \SplitArgument __cmd_split_argument:nnn
\TrimSpaces 3003 \cs_new_eq:NN \SplitList __cmd_split_list:nn
3004 \cs_new_eq:NN \TrimSpaces __cmd_trim_spaces:n
(End of definition for \ReverseBoolean and others.)

\ProcessList To support \SplitList.
3005 \cs_new_eq:NN \ProcessList \tl_map_function:nN

```

```
(End of definition for \ProcessList.)
```

Finally as promised, restore \\_\_kernel\_chk\_if\_free\_cs:N:

```
3006 <latexrelease>\cs_gset_eq:NN __kernel_chk_if_free_cs:N __cmd_chk_if_free_cs:N
3007 <latexrelease>\cs_undefine:N __cmd_chk_if_free_cs:N
```

```
3008 <latexrelease>
3009 <latexrelease>\IncludeInRelease{0000/00/00}{\ltcmd}%
3010 <latexrelease> {Document~command~parser}%
3011 <latexrelease>
3012 <latexrelease>\EndModuleRelease
3013 \ExplSyntaxOff
```

Now in `\textrm{latexrelease}` mode, redefine `\NewDocumentCommand` to not complain on commands already defined.

```
3014 <latexrelease>\@ifundefined{\ExplSyntaxOff}{\@textrm{latexrelease}\post\ltcmd}
3015 <latexrelease>\catcode`\^=\@textrm{latexrelease}\catcode@null\relax
3016 </2ekernel | \textrm{latexrelease}>
```

We need to stop DocStrip treating `@@` in a special way at this point.

```
3017 <@@=>
```

# File 08

## lthooks.dtx

### 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 L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> 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 L<sup>A</sup>T<sub>E</sub>X. 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 L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> interfaces

##### 2.1.1 Declaring hooks

With a few exceptions, hooks have to be declared before they can be used. The exceptions are the generic hooks for commands and environments (executed at `\begin` and `\end`), and the hooks run when loading files (see section 3.1).

---

`\NewHook \NewHook {⟨hook⟩}`

Creates a new `⟨hook⟩`. If this hook is declared within a package it is suggested that its 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 2.1.5. The string ?? can't be used as a hook name because it has a special significance as a placeholder in hook rules.

---

```
\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 2.1.5.

---

```
\NewMirroredHookPair \NewMirroredHookPair {\<hook-1>} {\<hook-2>}
```

---

A shorthand for `\NewHook{\<hook-1>} \NewReversedHook{\<hook-2>}`.

The `<hook>` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\NewHookWithArguments \NewHookWithArguments {\<hook>} {\<number>}
```

---

New: 2023-06-01

Creates a new `<hook>` whose code takes `<number>` arguments, and otherwise works exactly like `\NewHook`. Section 2.7 explains hooks with arguments.

The `<hook>` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\NewReversedHookWithArguments \NewReversedHookWithArguments {\<hook>} {\<number>}
```

---

New: 2023-06-01

Like `\NewReversedHook`, but creates a hook whose code takes `<number>` arguments. Section 2.7 explains hooks with arguments.

The `<hook>` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\NewMirroredHookPairWithArguments \NewMirroredHookPairWithArguments {\<hook-1>} {\<hook-2>} {\<number>}
```

---

New: 2023-06-01

A shorthand for `\NewHookWithArguments{\<hook-1>}{\<number>}`

`\NewReversedHookWithArguments{\<hook-2>}{\<number>}`. Section 2.7 explains hooks with arguments.

The `<hook>` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

## 2.1.2 Special declarations for generic hooks

The declarations here should normally not be used. They are available to provide support for special use cases mainly involving generic command hooks.

---

```
\DisableGenericHook \DisableGenericHook {\<hook>}
```

---

After this declaration<sup>3</sup> the `<hook>` is no longer usable: Any further attempt to add code to it will result in an error and any use, e.g., via `\UseHook`, will simply do nothing.

This is intended to be used with generic command hooks (see `lthcmdhooks-doc`) as depending on the definition of the command such generic hooks may be unusable. If that is known, a package developer can disable such hooks up front.

The `<hook>` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

<sup>3</sup>In the 2020/06 release this command was called `\DisableHook`, but that name was misleading as it shouldn't be used to disable non-generic hooks.

---

**\ActivateGenericHook** `\ActivateGenericHook {<hook>}`

This declaration activates a generic hook provided by a package/class (e.g., one used in code with `\UseHook` or `\UseOneTimeHook`) without it being explicitly declared with `\NewHook`). If the hook is already activated, this command does nothing.

Note that this command does not undo the effect of `\DisableGenericHook`. See section 2.6 for a discussion of when this declaration is appropriate.

### 2.1.3 Using hooks in code

---

**\UseHook** `\UseHook {<hook>}`

Execute the code stored in the `<hook>`.

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.

---

**\UseHookWithArguments** `\UseHookWithArguments {<hook>} {<number>} {<arg_1>} ... {<arg_n>}`

New: 2023-06-01

Execute the code stored in the `<hook>` and pass the arguments `{<arg_1>}` through `{<arg_n>}` to the `<hook>`. Otherwise, it works exactly like `\UseHook`. The `<number>` should be the number of arguments declared for the hook. If the hook is not declared, this command does nothing and it will remove `<number>` items from the input. Section 2.7 explains hooks with arguments.

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}`. From that point onwards, 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.

Using `\UseOneTimeHook` several times with the same `{<hook>}` means that it only executes the first time it is used. For example, if it is used in a command that can be called several times then the hook executes during only the *first* invocation of that command; this allows its use as an “initialization hook”.

Mixing `\UseHook` and `\UseOneTimeHook` for the same `{<hook>}` should be avoided, but if this is done then neither will execute after the first `\UseOneTimeHook`.

The `<hook>` *cannot* be specified using the dot-syntax. A leading `.` is treated literally. See section 2.1.5 for details.

---

```
\UseOneTimeHookWithArguments \UseOneTimeHookWithArguments {\hook} {\number} {\arg_1} ... {\arg_n}
```

---

New: 2023-06-01

Works exactly like `\UseOneTimeHook`, but passes arguments  $\{\arg_1\}$  through  $\{\arg_n\}$  to the  $\langle\text{hook}\rangle$ . The  $\langle\text{number}\rangle$  should be the number of arguments declared for the hook. If the hook is not declared, this command does nothing and it will remove  $\langle\text{number}\rangle$  items from the input.

It should be noted that after a one-time hook is used, it is no longer possible to use `\AddToHookWithArguments` or similar with that hook. `\AddToHook` continues to work as normal. Section 2.7 explains hooks with arguments.

The  $\langle\text{hook}\rangle$  *cannot* be specified using the dot-syntax. A leading `.` is treated literally. See section 2.1.5 for details.

#### 2.1.4 Updating code for hooks

---

```
\AddToHook \AddToHook {\hook} [|\label|] {\code}
```

---

Adds  $\langle\text{code}\rangle$  to the  $\langle\text{hook}\rangle$  labeled by  $\langle\text{label}\rangle$ . When the optional argument  $\langle\text{label}\rangle$  is not provided, the `\default_label` is used (see section 2.1.5). If `\AddToHook` is used in a package/class, the `\default_label` is the package/class name, otherwise it is `top-level` (the `top-level` label is treated differently: see section 2.1.6).

If there already exists code under the  $\langle\text{label}\rangle$  then the new  $\langle\text{code}\rangle$  is appended to the existing one (even if this is a reversed hook). If you want to replace existing code under the  $\langle\text{label}\rangle$ , first apply `\RemoveFromHook`.

The hook doesn't have to exist for code to be added to it. However, if it is not declared, then obviously the added  $\langle\text{code}\rangle$  will never be executed. This allows for hooks to work regardless of package loading order and enables packages to add to hooks from other packages without worrying whether they are actually used in the current document. See section 2.1.8.

The  $\langle\text{hook}\rangle$  and  $\langle\text{label}\rangle$  can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\AddToHookWithArguments \AddToHookWithArguments {\hook} [|\label|] {\code}
```

---

New: 2023-06-01

Works exactly like `\AddToHook`, except that the  $\langle\text{code}\rangle$  can access the arguments passed to the hook using `#1`, `#2`, ..., `#n` (up to the number of arguments declared for the hook). If the  $\langle\text{code}\rangle$  should contain *parameter tokens* (`#`) that are not supposed to be understood as the arguments of the hook, such tokens should be doubled. For example, with `\AddToHook` one can write:

```
\AddToHook{myhook}{\def\foo#1{Hello, #1!}}
```

but to achieve the same with `\AddToHookWithArguments`, one should write:

```
\AddToHookWithArguments{myhook}{\def\foo##1{Hello, ##1!}}
```

because in the latter case, `#1` refers to the first argument of the hook `myhook`. Section 2.7 explains hooks with arguments.

The  $\langle\text{hook}\rangle$  and  $\langle\text{label}\rangle$  can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\RemoveFromHook \RemoveFromHook {\hook} [{label}]
```

---

Removes any code labeled by `{label}` from the `{hook}`. When the optional argument `{label}` is not provided, the `{default label}` is used (see section 2.1.5).

If there is no code under the `{label}` in the `{hook}`, or if the `{hook}` does not exist, a warning is issued when you attempt to `\RemoveFromHook`, and the command is ignored. `\RemoveFromHook` should be used only when you know exactly what labels are in a hook. Typically this will be when some code gets added to a hook by a package, then later this code is removed by that same package. If you want to prevent the execution of code from another package, use the `voids` rule instead (see section 2.1.7).

If the optional `{label}` argument is `*`, then all code chunks are removed. This is rather dangerous as it may well drop code from other packages (that one may not know about); it should therefore not be used in packages but only in document preambles!

The `{hook}` and `{label}` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

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 two 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 this 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 of the hook happens after the end of that 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.<sup>4</sup>

If this declaration is used with a one-time hook then the code is only ever used if the declaration comes before the hook's invocation. This is because, in contrast to \AddToHook, the code in this declaration is not executed immediately in the case when the invocation of the hook has already happened—in other words, this code will truly execute only on the next invocation of the hook (and in the case of a one-time hook there is no such “next invocation”). This gives you a choice: should my code execute always, or should it execute only at the point where the one-time hook is used (and not at all if this is impossible)? For both of these possibilities there are use cases.

It is possible to nest this declaration 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 2.1.8.

The *hook* can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

**\AddToHookNextWithArguments** \AddToHookNextWithArguments {*hook*} {*code*}

New: 2023-06-01

Works exactly like \AddToHookNext, but the *code* can contain references to the arguments of the *hook* as described for \AddToHookWithArguments above. Section 2.7 explains hooks with arguments.

The *hook* can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

**\ClearHookNext** \ClearHookNext {*hook*}

Normally \AddToHookNext is only used when you know precisely where it will apply and why you want some extra code at that point. However, there are a few use cases in which such a declaration needs to be canceled, for example, when discarding a page with \DiscardShipoutBox (but even then not always), and in such situations \ClearHookNext can be used.

### 2.1.5 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*.

---

<sup>4</sup>There is no mechanism to reorder such code chunks (or delete them).

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` and `\IfHookEmptyTF` (and their `expl3` interfaces `\hook_use:n`, `\hook_use_once:n` and `\hook_if_empty: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`, `\UseOneTimeHook`, and `\IfHookEmptyTF` 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 the `\hook` name or the `\label` consist just of a single dot (.), or starts with a dot followed by a slash (./) then the dot denotes the `\default_label` (usually the current package or class name—see `\SetDefaultHookLabel`). 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/foo.tex/after}{code}
```

are equivalent to:

```
\NewHook {mymodule/hook}
\AddToHook {mymodule/hook}[mymodule]{code}
\AddToHook {mymodule/hook}[mymodule/sub]{code}
\DeclareHookRule{begindocument}{mymodule}{before}{babel}
\AddToHook {file/foo.tex/after}{code} % unchanged
```

The `\default_label` is automatically set equal to the name of the current package or class at the time the package is loaded. If 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 may have exceptions—see `\PushDefaultHookLabel`.

This syntax is available in all `\label` arguments and most `\hook` arguments, both in the `LaTeX2 $\varepsilon$`  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

**Important:**

The dot-syntax is not available with `\UseHook` and some other commands that are typically used within code!

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`), 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 the code in logical parts, but still use the main package name as the `<label>`, then the `<default label>` can be set using `\PushDefaultHookLabel{...} ... \PopDefaultHookLabel` or `\SetDefaultHookLabel{...}`.

---

```
\PushDefaultHookLabel \PushDefaultHookLabel {<default label>}
 <code>
\PopDefaultHookLabel
```

---

`\PushDefaultHookLabel` sets the current `<default label>` to be used in `<label>` arguments, or when replacing a leading “.” (see above). `\PopDefaultHookLabel` reverts the `<default label>` to its previous value.

Inside a package or class, the `<default label>` is equal to the package or class name, unless explicitly changed. Everywhere else, the `<default label>` is **top-level** (see section 2.1.6) unless explicitly changed.

The effect of `\PushDefaultHookLabel` holds until the next `\PopDefaultHookLabel`. `\usepackage` (and `\RequirePackage` and `\documentclass`) internally use

```
\PushDefaultHookLabel{<package name>}
 <package code>
\PopDefaultHookLabel
```

to set the `<default label>` for the package or class file. Inside the `<package code>` the `<default label>` can also be changed with `\SetDefaultHookLabel`. `\input` and other file input-related commands from the L<sup>A</sup>T<sub>E</sub>X kernel do not use `\PushDefaultHookLabel`, so code within files loaded by these commands does *not* get a dedicated `<label>`! (that is, the `<default label>` is the current active one when the file was loaded.)

Packages that provide their own package-like interfaces (TikZ's `\usetikzlibrary`, for example) can use `\PushDefaultHookLabel` and `\PopDefaultHookLabel` to set dedicated labels and to emulate `\usepackage`-like hook behavior within those contexts.

The **top-level** label is treated differently, and is reserved to the user document, so it is not allowed to change the `<default label>` to **top-level**.

---

```
\SetDefaultHookLabel \SetDefaultHookLabel {\defaultlabel}
```

---

Similarly to `\PushDefaultHookLabel`, sets the current `\defaultlabel` to be used in `\label` arguments, or when replacing a leading “.”. The effect holds until the label is changed again or until the next `\PopDefaultHookLabel`. The difference between `\PushDefaultHookLabel` and `\SetDefaultHookLabel` is that the latter does not save the current `\defaultlabel`.

This command is useful when a large package is composed of several smaller packages, but all should have the same `\label`, so `\SetDefaultHookLabel` can be used at the beginning of each package file to set the correct label.

`\SetDefaultHookLabel` is not allowed in the main document, where the `\defaultlabel` is **top-level** and there is no `\PopDefaultHookLabel` to end its effect. It is also not allowed to change the `\defaultlabel` to **top-level**.

### 2.1.6 The top-level label

The **top-level** label, assigned to code added from the main document, is different from other labels. Code added to hooks (usually `\AtBeginDocument`) in the preamble is almost always to change something defined by a package, so it should go at the very end of the hook.

Therefore, code added in the **top-level** is always executed at the end of the hook, regardless of where it was declared. If the hook is reversed (see `\NewReversedHook`), the **top-level** chunk is executed at the very beginning instead.

Rules regarding **top-level** have no effect: if a user wants to have a specific set of rules for a code chunk, they should use a different label to said code chunk, and provide a rule for that label instead.

The **top-level** label is exclusive for the user, so trying to add code with that label from a package results in an error.

### 2.1.7 Defining relations between hook code

The default assumption is that code added to hooks by different packages are independent and the order in which they are executed is irrelevant. While this is true in many cases it is obviously false in 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  $\langle\text{label1}\rangle$  and  $\langle\text{label2}\rangle$  for a given  $\langle\text{hook}\rangle$ . If  $\langle\text{hook}\rangle$  is ?? this defines a default relation for all hooks that use the two labels, i.e., that have chunks of code labeled with  $\langle\text{label1}\rangle$  and  $\langle\text{label2}\rangle$ .

Currently, the supported relations are the following:

**before** or < Code for  $\langle\text{label1}\rangle$  comes before code for  $\langle\text{label2}\rangle$ .

**after** or > Code for  $\langle\text{label1}\rangle$  comes after code for  $\langle\text{label2}\rangle$ .

**incompatible-warning** Only code for either  $\langle\text{label1}\rangle$  or  $\langle\text{label2}\rangle$  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-warning** but instead of a warning a L<sup>A</sup>T<sub>E</sub>X error is raised, and the code for both labels are dropped from that hook until the conflict is resolved.

**voids** Code for  $\langle\text{label1}\rangle$  overwrites code for  $\langle\text{label2}\rangle$ . More precisely, code for  $\langle\text{label2}\rangle$  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  $\langle\text{label1}\rangle$  and  $\langle\text{label2}\rangle$  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 declaration. In all cases rules specific to a given hook take precedence over default rules that use ?? as the  $\langle\text{hook}\rangle$ .

If a default rule is applied, it is done before reversing the label order in a reversed hook, e.g., **before** in a default rule effectively becomes **after** in such a hook. In contrast, a rule for a specific hook is always applied to the state after any reversal (i.e., the state you see when using `\ShowHook` on that hook).

The  $\langle\text{hook}\rangle$  and  $\langle\text{label}\rangle$  can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\ClearHookRule \ClearHookRule {⟨hook⟩} {⟨label1⟩} {⟨label2⟩}
```

Syntactic sugar for saying that  $\langle\text{label1}\rangle$  and  $\langle\text{label2}\rangle$  are unrelated for the given  $\langle\text{hook}\rangle$ .

---

```
\DeclareDefaultHookRule \DeclareDefaultHookRule {\langle label1\rangle} {\langle relation\rangle} {\langle label2\rangle}
```

---

This sets up a relation between `\langle label1\rangle` and `\langle label2\rangle` 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.)

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. The rationale is that in hook pairs (in which the ordering in one is reversed) default rules have to be reversed too in nearly all scenarios. If this is not the case, a default rule can't be used or has to be overwritten with an explicit `\DeclareHookRule` for that specific hook.

Declaring default rules is only supported in the document preamble.<sup>5</sup>

The `\langle label\rangle` can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

### 2.1.8 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: a hook may exist or not, and independently it may or may not be empty. This means that even a hook that doesn't exist may be non-empty and it can also be disabled.

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.

Given that code or rules can be added to a hook even if it doesn't physically exist yet, means that a querying its existence has no real use case (in contrast to other variables that can only be update if they have already been declared). For that reason only the test for emptiness has a public interface.

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. Generic hooks such as `file` and `env` hooks are automatically declared when code is added to them.

---

<sup>5</sup>Trying to do so, e.g., via `\DeclareHookRule` with `??` has bad side-effects and is not supported (though not explicitly caught for performance reasons).

---

\IfHookEmptyTF \* \IfHookEmptyTF {\hook} {\truecode} {\falsecode}  
\IfHookEmptyT \* Tests if the `\hook` is empty (*i.e.*, no code was added to it using either `\AddToHook` or `\AddToHookNext`) or such code was removed again (via `\RemoveFromHook`), and branches to either `\truecode` or `\falsecode` depending on the result.  
The `\hook` cannot be specified using the dot-syntax. A leading `.` is treated literally.

### 2.1.9 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 \ShowHook {\hook}  
\LogHook \LogHook {\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 2.1.5.

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 > Document-level (top-level) code (executed last):
7 > -> [code from 'top-level']
8 > Extra code for next invocation:
9 > -> [one-time code]
10 > Rules:
11 > foo|baz with relation >
12 > baz|bar with default relation <
13 > Execution order (after applying rules):
14 > baz, foo, bar.
```

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 added by the user in the main document (labeled **top-level**) in the format

```
Document-level (top-level) code (executed <first/last>):
-> <top-level code>
```

This code will be either the first or last code executed by the hook (**last** if the hook is normal, **first** if it is reversed). This chunk is not affected by rules and does not take part in sorting.

Line 9 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 11 and 12 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 this rule applies to *label-1* and *label-2* in *all* hooks, (unless overridden by a non-default relation).

Finally, line 14 lists the labels in the hook after sorting; that is, in the order they will be executed when the hook is used.

### 2.1.10 Debugging hook code

---

\DebugHooksOn \DebugHooksOff

---

\DebugHooksOff Turn the debugging of hook code on or off. This displays most changes made to the hook data structures. The output is rather coarse and not really intended for normal use.

## 2.2 L3 programming layer (exp13) interfaces

This is a quick summary of the LATEX3 programming interfaces for use with packages written in exp13. In contrast to the LATEX 2 $\varepsilon$  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 exp13 packages, but the choice is yours.

---

\hook\_new:n \hook\_new:n {*hook*}
\hook\_new\_reversed:n \hook\_new\_reversed:n {*hook*}
\hook\_new\_pair:nn \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 2.1.5.

---

```
\hook_new_with_args:nn \hook_new_with_args:nn {⟨hook⟩} {⟨number⟩}
\hook_new_reversed_with_args:nn \hook_new_reversed_with_args:nn {⟨hook⟩} {⟨number⟩}
\hook_new_pair_with_args:nnn \hook_new_pair_with_args:nnn {⟨hook-1⟩} {⟨hook-2⟩} {⟨number⟩}
```

---

New: 2023-06-01

Creates a new ⟨hook⟩ with normal or reverse ordering of code chunks, that takes ⟨number⟩ arguments from the input stream when it is used. \hook\_new\_pair\_with\_args: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 2.1.5.

---

```
\hook_disable_generic:n \hook_disable_generic:n {⟨hook⟩}
```

---

Marks {⟨hook⟩} as disabled. Any further attempt to add code to it or declare it, will result in an error and any call to \hook\_use:n will simply do nothing.

This declaration is intended for use with generic hooks that are known not to work (see `lcmdhooks-doc`) if they receive code.

The ⟨hook⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.5.

---

```
\hook_activate_generic:n \hook_activate_generic:n {⟨hook⟩}
```

---

This is like \hook\_new:n but it does nothing if the hook was previously declared with \hook\_new:n. This declaration should be used only in special situations, e.g., when a command from another package needs to be altered and it is not clear whether a generic cmd hook (for that command) has been previously explicitly declared.

Normally \hook\_new:n should be used instead of this.

---

```
\hook_use:n \hook_use:n {⟨hook⟩}
```

---

```
\hook_use:nnw \hook_use:nnw {⟨hook⟩} {⟨number⟩} {⟨arg₁⟩} ... {⟨argₙ⟩}
```

---

New: 2023-06-01 Executes the {⟨hook⟩} code followed (if set up) by the code for next invocation only, then empties that next invocation code. \hook\_use:nnw should be used for hooks declared with arguments, and should be followed by as many brace groups as the declared number of arguments. The ⟨number⟩ should be the number of arguments declared for the hook. If the hook is not declared, this command does nothing and it will remove ⟨number⟩ items from the input.

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

---

```
\hook_use_once:n \hook_use_once:n {⟨hook⟩}
```

---

```
\hook_use_once:nnw \hook_use_once:nnw {⟨hook⟩} {⟨number⟩} {⟨arg₁⟩} ... {⟨argₙ⟩}
```

---

New: 2023-06-01 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. \hook\_use\_once:nnw should be used for hooks declared with arguments, and should be followed by as many brace groups as the declared number of arguments. The ⟨number⟩ should be the number of arguments declared for the hook. If the hook is not declared, this command does nothing and it will remove ⟨number⟩ items from the input.

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}`

---

`\hook_gput_code_with_args:nnn` `\hook_gput_code_with_args:nnn {\hook} {\label} {\code}`

---

New: 2023-06-01

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 `\hook_gput_code_with_args:nnn` is used, the `\code` can access the arguments passed to `\hook_use:nnw` (or `\hook_use_once:nnw`) with #1, #2, ..., #n (up to the number of arguments declared for the hook). In that case, if an actual parameter token should be added to the code, it should be doubled.

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 2.1.5.

---

`\hook_gput_next_code:nn`

`\hook_gput_next_code:nn {\hook} {\code}`

---

`\hook_gput_next_code_with_args:nn` `\hook_gput_next_code_with_args:nn {\hook} {\code}`

---

New: 2023-06-01

Adds a chunk of `\code` for use only in the next invocation of the `\hook`. Once used it is gone.

If `\hook_gput_next_code_with_args:nn` is used, the `\code` can access the arguments passed to `\hook_use:nnw` (or `\hook_use_once:nnw`) with #1, #2, ..., #n (up to the number of arguments declared for the hook). In that case, if an actual parameter token should be added to the code, it should be doubled.

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 2.1.5.

---

`\hook_gclear_next_code:n` `\hook_gclear_next_code:n {\hook}`

Undo any earlier `\hook_gput_next_code:nn`.

---

`\hook_gremove_code:nn` `\hook_gremove_code:nn {\hook} {\label}`

Removes any code for `\hook` labeled `\label`.

If there is no code under the `\label` in the `\hook`, or if the `\hook` does not exist, a warning is issued when you attempt to use `\hook_gremove_code:nn`, and the command is ignored.

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 2.1.5.

---

```
\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 2.1.5. 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:nTF {<hook>} {<true code>} {<false code>}
```

---

`\hook_if_empty:nTF *` 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_show:n \hook_show:n {<hook>}
```

---

```
\hook_log:n \hook_log: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 2.1.5.

---

```
\hook_debug_on: \hook_debug_on:
```

---

`\hook_debug_off:` 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}
> Document-level (top-level) code (executed last):
> ---
> Extra code for 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, for 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}
> Document-level (top-level) code (executed last):
> ---
> Extra code for next invocation:
> ---
> Rules:
> packageB|packageA with relation >
> 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}{packageC}
```

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<sup>6</sup>, 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 another `package-too` makes the quotes also in blue and therefore adds:

```
\usepackage{color}
\AddToHook{env/quote/before}[package-too]{\begin{color}{blue}}
\AddToHook{env/quote/after} [package-too]{\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, package-too`

(or vice versa) and as a result, would get:

```
\begin{itshape}\begin{color}{blue} ...
\end{itshape}\end{color}
```

and an error message saying that `\begin{color}` was 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}
> package-too -> \end {color}
> Document-level (top-level) code (executed first):
> ---
> Extra code for next invocation:
> ---
> Rules:
> ---
> Execution order (after reversal):
> package-too, package-1.
```

If there is a matching default rule (done with `\DeclareDefaultHookRule` or with `??` for the hook name) then this default rule is applied before the reversal so that the order in the reversed hook mirrors the one in the normal hook. However, all rules specific to a hook happen always after the reversal of the execution order, 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.

---

<sup>6</sup>There are simpler ways to achieve the same effect.

## 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 `\langle code \rangle` immediately.

This has some consequences one needs to be aware of:

- If `\langle code \rangle` 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 `\langle code \rangle` will never be executed.
- In contrast if that happens with a one-time hook the `\langle code \rangle` is executed immediately.

In particular this means that construct such as

```
\AddToHook{myhook}
{ \langle code-1 \rangle \AddToHook{myhook}{\langle code-2 \rangle} \langle code-3 \rangle }
```

works for one-time hooks<sup>7</sup> (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 `\langle code-1 \rangle`,
- then execute `\AddToHook{myhook}{code-2}` which adds the code chunk `\langle code-2 \rangle` to the hook for use on the next invocation,
- and finally execute `\langle code-3 \rangle`.

The second time `\UseHook` is called it would execute the above and in addition `\langle code-2 \rangle` as that was added as a code chunk to the hook in the meantime. So each time the hook is used another copy of `\langle code-2 \rangle` is added and so that code chunk is executed `\langle \# of invocations \rangle - 1` times.

## 2.6 Generic hooks provided by packages

The hook management system also implements a category of hooks that are called “Generic Hooks”. Normally a hook has to be explicitly declared before it can be used in code. This ensures that different packages are not using the same hook name for unrelated purposes—something that would result in absolute chaos. However, there are a number of “standard” hooks where it is unreasonable to declare them beforehand, e.g., each and every command has (in theory) an associated `before` and `after` hook. In such cases, i.e., for command, environment or file hooks, they can be used simply by adding code to them with `\AddToHook`. For more specialized generic hooks, e.g., those provided

---

<sup>7</sup>This is sometimes used with `\AtBeginDocument` which is why it is supported.

by `babel`, you have to additionally enable them with `\ActivateGenericHook` as explained below.

The generic hooks provided by L<sup>A</sup>T<sub>E</sub>X are those for `cmd`, `env`, `file`, `include`, `package`, and `class`, and all these are available out of the box: you only have to use `\AddToHook` to add code to them, but you don't have to add `\UseHook` or `\UseOneTimeHook` to your code, because this is already done for you (or, in the case of `cmd` hooks, the command's code is patched at `\begin{document}`, if necessary).

However, if you want to provide further generic hooks in your own code, the situation is slightly different. To do this you should use `\UseHook` or `\UseOneTimeHook`, but *without declaring the hook* with `\NewHook`. As mentioned earlier, a call to `\UseHook` with an undeclared hook name does nothing. So as an additional setup step, you need to explicitly activate your generic hook. Note that a generic hook produced in this way is always a normal hook.

For a truly generic hook, with a variable part in the hook name, such upfront activation would be difficult or impossible, because you typically do not know what kind of variable parts may come up in real documents.

For example, `babel` provides hooks such as `babel/(language)/afterextras`. However, language support in `babel` is often done through external language packages. Thus doing the activation for all languages inside the core `babel` code is not a viable approach. Instead it needs to be done by each language package (or by the user who wants to use a particular hook).

Because the hooks are not declared with `\NewHook` their names should be carefully chosen to ensure that they are (likely to be) unique. Best practice is to include the package or command name, as was done in the `babel` example above.

Generic hooks defined in this way are always normal hooks (i.e., you can't implement reversed hooks this way). This is a deliberate limitation, because it speeds up the processing considerably.

## 2.7 Hooks with arguments

Sometimes it is necessary to pass contextual information to a hook, and, for one reason or another, it is not feasible to store such information in macros. To serve this purpose, hooks can be declared with arguments, so that the programmer can pass along the data necessary for the code in the hook to function properly.

A hook with arguments works mostly like a regular hook, and most commands that work for regular hooks, also work for hooks that take arguments. The differences are when the hook is declared (`\NewHookWithArguments` is used instead of `\NewHook`), then code can be added with both `\AddToHook` and `\AddToHookWithArguments`, and when the hook is used (`\UseHookWithArguments` instead of `\UseHook`).

A hook with arguments must be declared as such (before it is first used, as all regular hooks) using `\NewHookWithArguments{(hook)}{(number)}`. All code added to that hook can then use #1 to access the first argument, #2 to access the second, and so forth up to the number of arguments declared. However, it is still possible to add code with references to the arguments of a hook that was not yet declared (we will discuss that later). At their core, hooks are macros, so T<sub>E</sub>X's limit of 9 arguments applies, and a low-level T<sub>E</sub>X error is raised if you try to reference an argument number that doesn't exist.

To use a hook with arguments, just write `\UseHookWithArguments{⟨hook⟩}{⟨number⟩}` followed by a braced list of the arguments. For example, if the hook `test` takes three arguments, write:

```
\UseHookWithArguments{test}{3}{arg-1}{arg-2}{arg-3}
```

then, in the `⟨code⟩` of the hook, all instances of `#1` will be replaced by `arg-1`, `#2` by `arg-2` and so on. If, at the point of usage, the programmer provides more arguments than the hook is declared to take, the excess arguments are simply ignored by the hook. Behaviour is unpredictable<sup>8</sup> if too few arguments are provided. If the hook isn't declared, `⟨number⟩` arguments are removed from the input stream.

Adding code to a hook with arguments can be done with `\AddToHookWithArguments` as well as with the regular `\AddToHook`, to achieve different outcomes. The main difference when it comes to adding code to a hook, in this case, is firstly the possibility of accessing a hook's arguments, of course, and second, how parameter tokens (`#6`) are treated.

Using `\AddToHook` in a hook that takes arguments will work as it does for all other hooks. This allows a package developer to add arguments to a hook that otherwise had none without having to worry about compatibility. This means that, for example:

```
\AddToHook{test}{\def\foo#1{Hello, #1!}}
```

will define the same macro `\foo` regardless if the hook `test` takes arguments or not.

Using `\AddToHookWithArguments` allows the `⟨code⟩` added to access the arguments of the hook with `#1`, `#2`, and so forth, up to the number of the arguments declared in the hook. This means that if one wants to add a `#6` to the `⟨code⟩` that token must be doubled in the input. The same definition from above, using `\AddToHookWithArguments`, needs to be rewritten:

```
\AddToHookWithArguments{test}{\def\foo##1{Hello, ##1!}}
```

Extending the above example to use the hook arguments, we could rewrite something like (now from declaration to usage, to get the whole picture):

```
\NewHookWithArguments{test}{1}
\AddToHookWithArguments{test}{%
 \typeout{Defining foo with "#1"}
 \def\foo##1{Hello, ##1! Some text after: #1}%
}
\UseHook{test}{Howdy!}
\ShowCommand\foo
```

Running the code above prints in the terminal:

```
Defining foo with "Howdy!"
> \foo=macro:
#1->Hello, #1! Some text after: Howdy!.
```

---

<sup>8</sup>The hook *will* take the declared number of arguments, and what will happen depends on what was grabbed, and what the hook code does with its arguments.

Note how `##1` in the call to `\AddToHookWithArguments` became `#1`, and the `#1` was replaced by the argument passed to the hook. Should the hook be used again, with a different argument, the definition would naturally change.

It is possible to add code referencing a hook's arguments before such hook is declared and the number of hooks is fixed. However, if some code is added to the hook, that references more arguments than will be declared for the hook, there will be a low-level TeX error about an “Illegal parameter number” at the time the hook is declared, which will be hard to track down because at that point TeX can't know whence the offending code came from. Thus it is important that package writers explicitly document how many arguments (if any) each hook can take, so users of those packages know how many arguments can be referenced, and equally important, what each argument means.

## 2.8 Private L<sup>A</sup>T<sub>E</sub>X kernel hooks

There are a few places where it is absolutely essential for L<sup>A</sup>T<sub>E</sub>X 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 though the result is predetermined. Furthermore it forces package writers to unnecessarily add such rules if they add further code to the hook (or break L<sup>A</sup>T<sub>E</sub>X).

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@<hook>` or `\@kernel@after@<hook>`. 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.<sup>9</sup>

## 2.9 Legacy L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> interfaces

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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 L<sup>A</sup>T<sub>E</sub>X and more will follow. See the next section for what is already available.

---

<sup>9</sup>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` (see section 2.1.6) if done outside of a package or class or with the package/class name if called inside such a file (see section 2.1.5).

This way one can add 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 another package's 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 3.2), 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.

The few hooks that existed previously in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> used internally commands such as `\@begindocumenthook` and packages sometimes augmented them directly rather than working through `\AtBeginDocument`. 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 L<sup>A</sup>T<sub>E</sub>X we will turn this legacy support off, as it does unnecessary slow down the processing.

### 3 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> commands and environments augmented by hooks

In this section we describe the standard hooks that are now offered by L<sup>A</sup>T<sub>E</sub>X, or give pointers to other documents in which they are described. This section will grow over time (and perhaps eventually move to usrguide3).

#### 3.1 Generic hooks

As stated earlier, with the exception of generic hooks, all hooks must be declared with `\NewHook` before they can be used. All generic hooks have names of the form “⟨type⟩/⟨name⟩/⟨position⟩”, where ⟨type⟩ is from the predefined list shown below, and ⟨name⟩ is the variable part whose meaning will depend on the ⟨type⟩. The last component, ⟨position⟩, has more complex possibilities: it can always be `before` or `after`; for `env` hooks, it can also be `begin` or `end`; and for `include` hooks it can also be `end`. Each specific hook is documented below, or in `ltcmdhooks-doc.pdf` or `ltfilehook-doc.pdf`.

The generic hooks provided by L<sup>A</sup>T<sub>E</sub>X belong to one of the six types:

**env** Hooks executed before and after environments –  $\langle \text{name} \rangle$  is the name of the environment, and available values for  $\langle \text{position} \rangle$  are **before**, **begin**, **end**, and **after**;

**cmd** Hooks added to and executed before and after commands –  $\langle \text{name} \rangle$  is the name of the command, and available values for  $\langle \text{position} \rangle$  are **before** and **after**;

**file** Hooks executed before and after reading a file –  $\langle \text{name} \rangle$  is the name of the file (with extension), and available values for  $\langle \text{position} \rangle$  are **before** and **after**;

**package** Hooks executed before and after loading packages –  $\langle \text{name} \rangle$  is the name of the package, and available values for  $\langle \text{position} \rangle$  are **before** and **after**;

**class** Hooks executed before and after loading classes –  $\langle \text{name} \rangle$  is the name of the class, and available values for  $\langle \text{position} \rangle$  are **before** and **after**;

**include** Hooks executed before and after \included files –  $\langle \text{name} \rangle$  is the name of the included file (without the .tex extension), and available values for  $\langle \text{position} \rangle$  are **before**, **end**, and **after**.

Each of the hooks above are detailed in the following sections and in linked documentation.

### 3.1.1 Generic hooks for all environments

Every environment  $\langle \text{env} \rangle$  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 third argument of \NewDocumentEnvironment and 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 forth argument of \NewDocumentEnvironment and 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.

Given that these generic hook names involve / as part of their name they would not work if one tries to define an environment using a name that involves a /.<sup>10</sup>

Generic environment hooks are never one-time hooks even with environments that are supposed to appear only once in a document.<sup>11</sup> In contrast to other hooks there is also no need to declare them using \NewHook.

---

<sup>10</sup>Officially, LATEX names for environments should only consist of a sequence of letters, numbers, and the character \*, i.e., this is not a new restriction.

<sup>11</sup>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.

The hooks are only executed if `\begin{env}` and `\end{env}` is used. If the environment code is executed via low-level calls to `\langle env` and `\rangle 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.

Largely for compatibility with existing packages, the following four commands are also available to set the environment hooks; but for new packages we recommend directly using the hook names and `\AddToHook`.

---

**\BeforeBeginEnvironment** `\BeforeBeginEnvironment [<label>] {<env>} {<code>}`

This declaration adds to the `env/<env>/before` hook using the `<label>`. If `<label>` is not given, the `<default label>` is used (see section [2.1.5](#)).

---

**\AtBeginEnvironment** `\AtBeginEnvironment [<label>] {<env>} {<code>}`

This is like `\BeforeBeginEnvironment` but it adds to the `env/<env>/begin` hook.

---

**\AtEndEnvironment** `\AtEndEnvironment [<label>] {<env>} {<code>}`

This is like `\BeforeBeginEnvironment` but it adds to the `env/<env>/end` hook.

---

**\AfterEndEnvironment** `\AfterEndEnvironment [<label>] {<env>} {<code>}`

This is like `\BeforeBeginEnvironment` but it adds to the `env/<env>/after` hook.

### 3.1.2 Generic hooks for commands

Similar to environments there are now (at least in theory) two generic hooks available for any L<sup>A</sup>T<sub>E</sub>X command. These are

**cmd/<name>/before** This hook is executed at the very start of the command execution.

**cmd/<name>/after** This hook is executed at the very end of the command body. It is implemented as a reversed hook.

In practice there are restrictions and especially the `after` hook works only with a subset of commands. Details about these restrictions are documented in `ltcmdhooks-doc.pdf` or with code in `ltcmdhooks-code.pdf`.

### 3.1.3 Generic hooks provided by file loading operations

There are several hooks added to L<sup>A</sup>T<sub>E</sub>X's process of loading file via its high-level interfaces such as `\input`, `\include`, `\usepackage`, `\RequirePackage`, etc. These are documented in `ltfilehook-doc.pdf` or with code in `ltfilehook-code.pdf`.

### 3.2 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):

**`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 by using `\AddToHook{begindocument}` or by using `\AtBeginDocument` and it is executed after the `.aux` file has been read 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 L<sup>A</sup>T<sub>E</sub>X'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.

### 3.3 Hooks provided by `\end{document}`

L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> has always provided `\AtEndDocument` to add code to the `\end{document}`, just in front of the code that is normally executed there. While this was a big improvement over the situation in L<sup>A</sup>T<sub>E</sub>X 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 `\end{document}`.

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, but since there will be no more pages you need to write to it using `\immediate\write`). 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 `\shipouts` is the last one,

$\text{\LaTeX}$  needs to be run several times, so initially it might get executed on the wrong page. See section 3.4 for where to find the details.

It is also possible to use the generic `env/document/end` hook which is executed 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  $\text{\LaTeX}$  has finished the document processing.

### 3.4 Hooks provided by `\shipout` operations

There are several hooks and mechanisms added to  $\text{\LaTeX}$ 's process of generating pages. These are documented in `ltshipout-doc.pdf` or with code in `ltshipout-code.pdf`.

### 3.5 Hooks provided for paragraphs

The paragraph processing has been augmented to include a number of internal and public hooks. These are documented in `ltpara-doc.pdf` or with code in `ltpara-code.pdf`.

### 3.6 Hooks provided in NFSS commands

In languages that need to support more than one script in parallel (and thus several sets of fonts, e.g., supporting both 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 commands have hooks to which such support can be added.

**rmfamily** After `\rmfamily` has done its initial checks and prepared a font series update, this hook is executed before `\selectfont`.

**sffamily** This is like the `rmfamily` hook, but for the `\sffamily` command.

**ttfamily** This is like the `rmfamily` hook, but for the `\ttfamily` command.

**normalfont** The `\normalfont` command resets the 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. The `bfseries/defaults` hook allows further adjustments to 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 they are in the `\mdseries` command.

**selectfont** This hook is executed inside `\selectfont`, after the current values for *encoding*, *family*, *series*, *shape*, and *size* are evaluated and the new font is selected (and if necessary loaded). After the hook has executed, NFSS will still do any updates necessary for a new *size* (such as changing the size of `\strut`) and any updates necessary to a change in *encoding*.

This hook is intended for use cases where, in parallel to a change in the main font, some other fonts need to be altered (e.g., in CJK processing where you may need to deal with several different alphabets).

### 3.7 Hook provided by the mark mechanism

See `ltmarks-doc.pdf` for details.

**insertmark** This hook allows for a special setup while `\InsertMark` inserts a mark. It is executed in group so local changes only apply to the mark being inserted.

## 4 The Implementation

```

1 <@@=hook>
2 <*2ekernel | latexrelease>
3 \ExplSyntaxOn
4 <latexrelease> \NewModuleRelease{2020/10/01}{lthooks}
5 <latexrelease> {The~hook~management~system}
```

### 4.1 Debugging

`\g__hook_debug_bool` Holds the current debugging state.

```
6 \bool_new:N \g__hook_debug_bool
```

(End of definition for `\g__hook_debug_bool`.)

```

\hook_debug_on: Turns debugging on and off by redefining __hook_debug:n.
\hook_debug_off: __hook_debug:n \use_none:n
__hook_debug:n \cs_new_eq:NN __hook_debug:n \use_none:n
__hook_debug_gset: \cs_new_protected:Npn \hook_debug_on:
 {
 \bool_gset_true:N \g__hook_debug_bool
 __hook_debug_gset:
 }
__hook_debug_gset: \cs_new_protected:Npn \hook_debug_off:
 {
 \bool_gset_false:N \g__hook_debug_bool
 __hook_debug_gset:
 }
__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 of definition for `\hook_debug_on:` and others. These functions are documented on page 209.)

## 4.2 Borrowing from internals of other kernel modules

\\_\\_hook\\_str\\_compare:nn Private copy of \\_\\_str\\_if\\_eq:nn  
23 \cs\_new\_eq:NN \\_\\_hook\\_str\\_compare:nn \\_\\_str\\_if\\_eq:nn  
(End of definition for \\_\\_hook\\_str\\_compare:nn.)

## 4.3 Declarations

\l\\_\\_hook\\_tmpa\\_bool Scratch boolean used throughout the package.

24 \bool\_new:N \l\\_\\_hook\\_tmpa\\_bool

(End of definition for \l\\_\\_hook\\_tmpa\\_bool.)

\l\\_\\_hook\\_return\\_tl Scratch variables used throughout the package.

25 \tl\_new:N \l\\_\\_hook\\_return\\_tl

26 \tl\_new:N \l\\_\\_hook\\_tmpa\\_tl

27 \tl\_new:N \l\\_\\_hook\\_tmpb\\_tl

(End of 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.

28 \seq\_new:N \g\\_\\_hook\\_all\\_seq

(End of definition for \g\\_\\_hook\\_all\\_seq.)

\l\\_\\_hook\\_cur\\_hook\\_tl Stores the name of the hook currently being sorted.

29 \tl\_new:N \l\\_\\_hook\\_cur\\_hook\\_tl

(End of 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.

30 \prop\_new:N \l\\_\\_hook\\_work\\_prop

(End of definition for \l\\_\\_hook\\_work\\_prop.)

\g\\_\\_hook\\_used\\_prop All hooks that receive code (for use in debugging display).

31 \prop\_new:N \g\\_\\_hook\\_used\\_prop

(End of 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.

32 \tl\_new:N \g\\_\\_hook\\_hook\\_curr\\_name\\_tl

33 \seq\_new:N \g\\_\\_hook\\_name\\_stack\\_seq

(End of definition for \g\\_\\_hook\\_hook\\_curr\\_name\\_tl and \g\\_\\_hook\\_name\\_stack\\_seq.)

\\_\\_hook\\_tmp:w Temporary macro for generic usage.

34 \cs\_new\_eq:NN \\_\\_hook\\_tmp:w ?

(End of definition for \\_\\_hook\\_tmp:w.)

|                                                  |                                                                                                                                                                                                                                                |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\c__hook_empty_tl</code>                   | An empty token list, and one containing nine parameters.                                                                                                                                                                                       |
|                                                  | <code>35 \tl_const:Nn \c__hook_empty_tl { }</code>                                                                                                                                                                                             |
|                                                  | <code>36 \tl_const:Nn \c__hook_nine_parameters_tl { #1#2#3#4#5#6#7#8#9 }</code>                                                                                                                                                                |
|                                                  | <i>(End of definition for \c__hook_empty_tl and \c__hook_nine_parameters_tl.)</i>                                                                                                                                                              |
| <code>\tl_gremove_once:Nx</code>                 | Some variants of <code>expl3</code> functions.                                                                                                                                                                                                 |
|                                                  | <i>FMi: should probably be moved to expl3</i>                                                                                                                                                                                                  |
|                                                  | <code>37 \cs_generate_variant:Nn \tl_gremove_once:Nn { Nx }</code>                                                                                                                                                                             |
|                                                  | <code>38 \cs_generate_variant:Nn \tl_show:n { x }</code>                                                                                                                                                                                       |
|                                                  | <code>39 \cs_generate_variant:Nn \tl_log:n { x }</code>                                                                                                                                                                                        |
|                                                  | <code>40 \cs_generate_variant:Nn \tl_set:Nn { Ne }</code>                                                                                                                                                                                      |
|                                                  | <code>41 \cs_generate_variant:Nn \cs_replacement_spec:N { c }</code>                                                                                                                                                                           |
|                                                  | <code>42 \cs_generate_variant:Nn \prop_put:Nnn { Nne }</code>                                                                                                                                                                                  |
|                                                  | <code>43 \cs_generate_variant:Nn \str_count:n { e }</code>                                                                                                                                                                                     |
|                                                  | <i>(End of definition for \tl_gremove_once:Nx and others.)</i>                                                                                                                                                                                 |
| <code>\s__hook_mark</code>                       | Scan mark used for delimited arguments.                                                                                                                                                                                                        |
|                                                  | <code>44 \scan_new:N \s__hook_mark</code>                                                                                                                                                                                                      |
|                                                  | <i>(End of definition for \s__hook_mark.)</i>                                                                                                                                                                                                  |
| <code>\_hook_use_none_delimit_by_s_mark:w</code> | Removes tokens until the next <code>\s__hook_mark</code> .                                                                                                                                                                                     |
| <code>\_hook_use_i_delimit_by_s_mark:nw</code>   | <code>45 \cs_new:Npn \_hook_use_none_delimit_by_s_mark:w #1 \s__hook_mark { }</code>                                                                                                                                                           |
|                                                  | <code>46 \cs_new:Npn \_hook_use_i_delimit_by_s_mark:nw #1 #2 \s__hook_mark {#1}</code>                                                                                                                                                         |
|                                                  | <i>(End of definition for \_hook_use_none_delimit_by_s_mark:w and \_hook_use_i_delimit_by_s_mark:nw.)</i>                                                                                                                                      |
| <code>\_hook_tl_set:cn</code>                    | Private copies of a few <code>expl3</code> functions. <code>\l3debug</code> will only add debugging to the public names, not to these copies, so we don't have to use <code>\debug_suspend:</code> and <code>\debug_resume:</code> everywhere. |
|                                                  | Functions like <code>\_hook_tl_set:Nn</code> have to be redefined, rather than copied because in <code>expl3</code> they use <code>\_kernel_tl_(g)set:Nx</code> , which is also patched by <code>\l3debug</code> .                             |
|                                                  | <code>47 \cs_new_protected:Npn \_hook_tl_set:cn #1#2</code>                                                                                                                                                                                    |
|                                                  | <code>48 { \cs_set_nopar:cp {#1} { \_kernel_exp_not:w {#2} } }</code>                                                                                                                                                                          |
|                                                  | <i>(End of definition for \_hook_tl_set:cn.)</i>                                                                                                                                                                                               |
| <code>\_hook_tl_gset:Nn</code>                   | Same as above.                                                                                                                                                                                                                                 |
|                                                  | <code>49 \cs_new_protected:Npn \_hook_tl_gset:Nn #1#2</code>                                                                                                                                                                                   |
|                                                  | <code>50 { \cs_gset_nopar:Npx #1 { \_kernel_exp_not:w {#2} } }</code>                                                                                                                                                                          |
|                                                  | <code>51 \cs_new_protected:Npn \_hook_tl_gset:Nx #1#2</code>                                                                                                                                                                                   |
|                                                  | <code>52 { \cs_gset_nopar:Npx #1 {#2} }</code>                                                                                                                                                                                                 |
|                                                  | <code>53 \cs_generate_variant:Nn \_hook_tl_gset:Nn { c, co }</code>                                                                                                                                                                            |
|                                                  | <code>54 \cs_generate_variant:Nn \_hook_tl_gset:Nx { c }</code>                                                                                                                                                                                |
|                                                  | <i>(End of definition for \_hook_tl_gset:Nn.)</i>                                                                                                                                                                                              |
| <code>\_hook_tl_gput_right:Nn</code>             | Same as above.                                                                                                                                                                                                                                 |
| <code>\_hook_tl_gput_right:Ne</code>             | <code>55 \cs_new_protected:Npn \_hook_tl_gput_right:Nn #1#2</code>                                                                                                                                                                             |
| <code>\_hook_tl_gput_right:cn</code>             | <code>56 { \_hook_tl_gset:Nx #1 { \_kernel_exp_not:w \exp_after:wN { #1 #2 } } }</code>                                                                                                                                                        |
|                                                  | <code>57 \cs_generate_variant:Nn \_hook_tl_gput_right:Nn { Ne, cn }</code>                                                                                                                                                                     |

(End of definition for `\_\_hook_tl_gput_right:Nn`.)

`\_\_hook_tl_gput_left:Nn` Same as above.

```
58 \cs_new_protected:Npn __hook_tl_gput_left:Nn #1#2
 59 {
 60 __hook_tl_gset:Nx #1
 61 { __kernel_exp_not:w {#2} __kernel_exp_not:w \exp_after:wN {#1} }
 62 }
```

(End of definition for `\_\_hook_tl_gput_left:Nn`.)

`\_\_hook_tl_gset_eq:NN` Same as above.

```
63 \cs_new_eq:NN __hook_tl_gset_eq:NN \tl_gset_eq:NN
```

(End of definition for `\_\_hook_tl_gset_eq:NN`.)

`\_\_hook_tl_gclear:N` Same as above.

```
64 \cs_new_protected:Npn __hook_tl_gclear:N #1
 65 { __hook_tl_gset_eq:NN #1 \c_empty_tl }
 66 \cs_generate_variant:Nn __hook_tl_gclear:N { c }
```

(End of definition for `\_\_hook_tl_gclear:N`.)

## 4.4 Providing new hooks

### 4.4.1 The data structures of a hook

`\g_@@_{hook}_code_prop` Hooks have a name (called `<hook>` in the description below) and for each hook we have `\@@_{hook}` to provide a number of data structures. These are

`\g_@@_{hook}_reversed_t1` `\g_@@_{hook}_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}_declared_t1` `\g_@@_{hook}_parameter_t1` `\g_@@_{hook}_rule_{label1}|{label2}_t1` A token list holding the relation between `<label1>` and `<label2>` in the `<hook>`. The `<labels>` are lexically (reverse) sorted to ensure that two labels always point to the same token list. For global rules, the `<hook>` name is `??`.

`\_\_hook_{hook}` 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}_reversed_t1` 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.

`\g_@@_{hook}_declared_t1` This token list serves as a marker for the hook being officially declared. Its existence is tested to raise an error in case another declaration is attempted.

`\c__hook_<hook>_parameter_t1` This token list stores the parameter text for a declared hook (its existence almost completely intersects the token list above), which is used for managing hooks with arguments.

`\__hook_toplevel_<hook>` This token list stores the code inserted in the hook from the user's document, in the `top-level` label. This label is special, and doesn't participate in sorting. Instead, all code is appended to it and executed after (or before, if the hook is reversed) the normal hook code, but before the `next` code chunk.

`\__hook_next_<hook>` 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_<hook>` token list. It is called `\__hook_next_<hook>` rather than `\__hook_next_<hook>` because otherwise a hook whose name is `next_<hook>` would clash with the next code-token list of the hook called `<hook>`.

#### 4.4.2 On the existence of hooks

A hook may be in different states of existence. Here we give an overview of the internal commands to set up hooks and explain how the different states are distinguished. The actual implementation then follows in subsequent sections.

One problem we have to solve is that we need to be able to add code to hooks (e.g., with `\AddToHook`) even if that code has not yet been declared. For example, one package needs to write into a hook of another package, but that package may not get loaded, or is loaded only later. Another problem is that most hooks, but not the generic hooks, require a declaration.

We therefore distinguish the following states for a hook, which are managed by four different tests: structure existence (`\__hook_if_structure_exist:nTF`), creation (`\__hook_if_usable:nTF`), declaration (`\__hook_if_declared:nTF`) and disabled or not (`\__hook_if_disabled:nTF`)

**not existing** Nothing is known about the hook so far. This state can be detected with `\__hook_if_structure_exist:nTF` (which uses the false branch).

In this state the hook can be declared, disabled, rules can be defined or code could be added to it, but it is not possible to use the hook (with `\UseHook`).

**basic data structure set up** A hook is this state when its basic data structure has been set up (using `\__hook_init_structure:n`). The data structure setup happens automatically when commands such as `\AddToHook` are used and the hook is at that point in state "not existing".

In this state the four tests give the following results:

```
__hook_if_structure_exist:nTF returns true.
 __hook_if_usable:nTF returns false.
 __hook_if_declared:nTF returns false.
 __hook_if_disabled:nTF returns false.
```

The allowed actions are the same as in the "not existing" state.

**declared** A hook is in this state it is not disabled and was explicitly declared (e.g., with `\NewHook`). In this case the four tests give the following results:

```
__hook_if_structure_exist:nTF returns true.
 __hook_if_usable:nTF returns true.
 __hook_if_declared:nTF returns true.
 __hook_if_disabled:nTF returns false.
```

**usable** A hook is in this state if it is not disabled, was not explicitly declared but nevertheless is allowed to be used (with `\UseHook` or `\hook_use:n`). This state is only possible for generic hooks as they do not need to be declared. Therefore such hooks move directly from state “not existing” to “usable” the moment a declaration such as `\AddToHook` wants to add to the hook data structure. In this state the tests give the following results:

```
__hook_if_structure_exist:nTF returns true.
 __hook_if_usable:nTF returns true.
 __hook_if_declared:nTF returns false.
 __hook_if_disabled:nTF returns false.
```

**disabled** A generic hook in any state is moved to this state when `\DisableGenericHook` is used. This changes the tests to give the following results:

```
__hook_if_structure_exist:nTF unchanged.
 __hook_if_usable:nTF returns false.
 __hook_if_declared:nTF returns true.
 __hook_if_disabled:nTF returns true.
```

The structure test is unchanged (if the hook was unknown before it is `false`, otherwise `true`). The usable test returns `false` so that any `\UseHook` will bypass the hook from now on. The declared test returns `true` so that any further `\NewHook` generates an error and the disabled test returns `true` so that `\AddToHook` can return an error.

*FMi: maybe it should do this only after begin document?*

#### 4.4.3 Setting hooks up

**\hook\_new:n**  
**\hook\_new\_with\_args:nn**

```
67 \hook_new:n
68 \hook_new_with_args:nn
69 \cs_new_protected:Npn \hook_new:n #1
70 { __hook_normalize_hook_args:Nn __hook_new:nn {#1} { 0 } }
71 \cs_new_protected:Npn \hook_new_with_args:nn #1 #2
72 { __hook_normalize_hook_args:Nn __hook_new:nn {#1} {#2} }
73 \cs_new_protected:Npn __hook_new:nn #1 #2
74 {
```

We check if the hook was already *explicitly* declared with `\hook_new:n`, and if it already exists we complain, otherwise set the “created” flag for the hook so that it errors next time `\hook_new:n` is used.

```

75 __hook_if_declared:nTF {#1}
76 { \msg_error:nnn { hooks } { exists } {#1} }
77 {
78 \tl_new:c { g__hook_#1_declared_tl }
79 \cs_undefine:c { __hook~#1 }
80 \cs_undefine:c { c__hook_#1_parameter_tl }
81 __hook_make_usable:nn {#1} {#2}

```

In case there is already code in a hook, but it’s undeclared, run `\__hook_update_hook_code:n` to make it ready to be executed (see test `lthooks-034`).

```

82 __hook_update_hook_code:n {#1}
83 }
84 }
85 \EndIncludeInRelease
86 \IncludeInRelease{2020/10/01}{\hook_new_with_args:nn}
87 {Hooks~with-args}
88 \cs_gset_protected:Npn \hook_new:n #1
89 { __hook_normalize_hook_args:Nn __hook_new:n {#1} }
90 \cs_undefine:N __hook_new:nn
91 \cs_gset_protected:Npn __hook_new:n #1
92 \{
93 __hook_if_declared:nTF {#1}
94 { \msg_error:nnn { hooks } { exists } {#1} }
95 \{
96 \tl_new:c { g__hook_#1_declared_tl }
97 __hook_make_usable:n {#1}
98 \}
99 \}
100 \cs_gset_protected:Npn \hook_new_with_args:nn #1 { }
101 \EndIncludeInRelease

```

(End of definition for `\hook_new:n`, `\hook_new_with_args:nn`, and `\__hook_new:nn`. These functions are documented on page [206](#).)

### `\__hook_make_usable:nn`

This initializes all hook data structures for the hook but if used on its own doesn’t mark the hook as declared (as `\hook_new:n` does, so a later `\hook_new:n` on that hook will not result in an error. This command is internally used by `\hook_gput_code:nnn` when adding code to a generic hook.

```

102 \IncludeInRelease{2023/06/01}{__hook_make_usable:nn}
103 {Hooks~with-args}
104 \cs_new_protected:Npn __hook_make_usable:nn #1 #2
105 {

```

Now we check if the hook’s data structure can be safely created without `expl3` raising errors, then we add the hook name to the list of all hooks and allocate the necessary data structures for the new hook, otherwise just do nothing.

```

106 __hook_if_usable:nF {#1}
107 {
108 \seq_gput_right:Nn \g__hook_all_seq {#1}

```

Here we'll define the `\c__hook_<hook>_parameter_tl` to hold a run of parameters up to the number of arguments of the hook (#2).

```

109 _kernel_cs_parm_from_arg_count:nNF
110 { \tl_const:cn { c__hook_#1_parameter_tl } } {#2}
111 {
112 \msg_error:nnn { hooks } { too-many-args } {#1} {#2}
113 \tl_const:cx { c__hook_#1_parameter_tl }
114 { \exp_not:V \c__hook_nine_parameters_tl }
115 }

```

After that, use `\_hook_normalise_cs_args:nn` to correct the number of parameters of the macros `\_hook_toplevel<hook>` and `\_hook_next<hook>`. We need to be able to add code with arguments to a hook without prior knowledge of the number of arguments of that hook, so `lthooks` assumes 9 until the hook is properly declared and the number of arguments is known. `\_hook_normalise_cs_args:nn` does the normalisation by using the `\c__hook_<hook>_parameter_tl` defined just above.

```

116 _hook_normalise_cs_args:nn { _toplevel } {#1}
117 _hook_normalise_cs_args:nn { _next } {#1}

```

This is only used by the actual code of the current hook, so declare it normally:

```
118 _hook_code_gset:nn {#1} { }
```

Now ensure that the base data structure for the hook exists:

```
119 _hook_init_structure:n {#1}
```

The call to `\_hook_normalise_code_pool:n` will correct any improper reference to arguments that don't exist in the hook, raising a low-level `TEX` error and doubling the offending parameter tokens. It has to be done after `\_hook_init_structure:n` because it operates on `\g__hook_<hook>_code_prop`.

```
120 _hook_normalise_code_pool:n {#1}
```

The `\g__hook_<hook>_labels_clist` holds the sorted list of labels (once it got sorted). This is used only for debugging. These are defined conditionally, in case `\_hook_make_usable:nn` is being used to redefine a hook.

```

121 \clist_if_exist:cF { g__hook_#1_labels_clist }
122 {
123 \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.

```

124 \tl_new:c { g__hook_#1_reversed_tl }
125 }

```

The above is all in L3 convention, but we also provide an interface to legacy `LATEX 2 $\varepsilon$`  hooks of the form `\@...hook`, e.g., `\begin{document}hook`. 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!**

```

126 _hook_include_legacy_code_chunk:n {#1}
127 }
128 }
129 \If@InRelease \EndIncludeInRelease

```

```

130 〈latexrelease〉\IncludeInRelease{2020/10/01}{__hook_make_usable:nn}
131 〈latexrelease〉 {Hooks~with~args}
132 〈latexrelease〉\cs_undefine:N __hook_make_usable:nn
133 〈latexrelease〉\cs_gset_protected:Npn __hook_make_usable:n #1
134 〈latexrelease〉 {
135 〈latexrelease〉 \tl_if_exist:cF { __hook~#1 }
136 〈latexrelease〉 {
137 〈latexrelease〉 \seq_gput_right:Nn \g__hook_all_seq {#1}
138 〈latexrelease〉 \tl_new:c { __hook~#1 }
139 〈latexrelease〉 __hook_init_structure:n {#1}
140 〈latexrelease〉 \clist_new:c { g__hook_#1_labels_clist }
141 〈latexrelease〉 \tl_new:c { g__hook_#1_reversed_tl }
142 〈latexrelease〉 __hook_include_legacy_code_chunk:n {#1}
143 〈latexrelease〉 }
144 〈latexrelease〉 }
145 〈latexrelease〉\EndIncludeInRelease

```

(End of definition for `\_\_hook\_make\_usable:nn`.)

`\_\_hook_init_structure:n` This function declares the basic data structures for a hook without explicit declaring the hook itself. This is needed to allow adding to undeclared hooks. Here it is unnecessary to check whether all variables exist, since all three are declared at the same time (either all of them exist, or none).

It creates the hook code pool (`\g__hook_<hook>_code_prop`) and the `top-level` and `next` token lists. A hook is initialized with `\_\_hook_init_structure:n` the first time anything is added to it. Initializing a hook just with `\_\_hook_init_structure:n` will not make it usable with `\hook_use:n`.

```

146 〈latexrelease〉\IncludeInRelease{2023/06/01}{__hook_init_structure:n}
147 〈latexrelease〉 {Hooks~with~args}
148 \cs_new_protected:Npn __hook_init_structure:n #1
149 {
150 __hook_if_structure_exist:nF {#1}
151 {
152 \prop_new:c { g__hook_#1_code_prop }
153 __hook_toplevel_gset:nn {#1} { }
154 __hook_next_gset:nn {#1} { }
155 }
156 }
157 〈latexrelease〉\EndIncludeInRelease
158 〈latexrelease〉\IncludeInRelease{2020/10/01}{__hook_init_structure:n}
159 〈latexrelease〉 {Hooks~with~args}
160 \cs_gset_protected:Npn __hook_init_structure:n #1
161 〈latexrelease〉 {
162 〈latexrelease〉 __hook_if_structure_exist:nF {#1}
163 〈latexrelease〉 {
164 〈latexrelease〉 \prop_new:c { g__hook_#1_code_prop }
165 〈latexrelease〉 \tl_new:c { __hook_toplevel~#1 }
166 〈latexrelease〉 \tl_new:c { __hook_next~#1 }
167 〈latexrelease〉 }
168 〈latexrelease〉 }
169 〈latexrelease〉\EndIncludeInRelease

```

(End of definition for `\_\_hook_init_structure: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.

```

__hook_new_reversed:nn 170 \IncludeInRelease{2023/06/01}{\hook_new_reversed_with_args:nn}
 {Hooks~with~args}
171 \cs_new_protected:Npn \hook_new_reversed:n #1
172 { __hook_normalize_hook_args:Nn __hook_new_reversed:nn {#1} { 0 } }
173 \cs_new_protected:Npn \hook_new_reversed_with_args:nn #1 #2
174 { __hook_normalize_hook_args:Nn __hook_new_reversed:nn {#1} {#2} }
175 \cs_new_protected:Npn __hook_new_reversed:nn #1 #2
176 {
177 __hook_if_declared:nTF {#1}
178 { \msg_error:nnn { hooks } { exists } {#1} }
179 {
180 __hook_new:nn {#1} {#2}
181 \tl_gset:cn { g__hook_#1_reversed_tl } { - }
182 }
183 }
184 }
185 \EndIncludeInRelease
186 \IncludeInRelease{2020/10/01}{\hook_new_reversed_with_args:nn}
 {Hooks~with~args}
187 \cs_gset_protected:Npn \hook_new_reversed:n #1
188 { __hook_normalize_hook_args:Nn __hook_new_reversed:n {#1} }
189 \cs_undefine:N __hook_new_reversed:nn
190 \cs_gset_protected:Npn __hook_new_reversed:n #1
191 \cs_gset_protected:Npn __hook_new_reversed:n #1
192 \cs_gset_protected:Npn __hook_new:n {#1}
193 \cs_gset_protected:Npn __hook_new_reversed:nn {#1}
194 \cs_gset_protected:Npn __hook_new_reversed_with_args:nn #1 #2 { }
195 \EndIncludeInRelease

```

(End of definition for `\hook_new_reversed:n`, `\hook_new_reversed_with_args:nn`, and `\__hook_new_reversed:nn`. These functions are documented on page 206.)

`\hook_new_pair:nn` A shorthand for declaring a normal and a (matching) reversed hook in one go.

```

\hook_new_pair_with_args:nnn 199 \IncludeInRelease{2023/06/01}{\hook_new_pair_with_args:nnn}
 {Hooks~with~args}
200 \cs_new_protected:Npn \hook_new_pair:nn #1#2
201 { __hook_normalize_hook_args:Nnn __hook_new_pair:nnn {#1} {#2} { 0 } }
202 \cs_new_protected:Npn \hook_new_pair_with_args:nnn #1#2#3
203 { __hook_normalize_hook_args:Nnn __hook_new_pair:nnn {#1} {#2} {#3} }
204 \cs_new_protected:Npn __hook_new_pair:nnn #1 #2 #3
205 {
206 __hook_if_declared:nTF {#1}
207 { \msg_error:nnn { hooks } { exists } {#1} }
208 {
209 __hook_if_declared:nTF {#2}
210 { \msg_error:nnn { hooks } { exists } {#2} }
211 {
212 __hook_new:nn {#1} {#3}
213 __hook_new_reversed:nn {#2} {#3}
214 }
215 }
216 }

```

```

217 }
218 \EndIncludeInRelease
219 \IncludeInRelease{2020/10/01}{\hook_new_pair_with_args:nnn}
220 {Hooks~with~args}
221 \cs_gset_protected:Npn \hook_new_pair:nn #1#2
222 {
223 \hook_new:n {#1}
224 \hook_new_reversed:n {#2}
225 }
226 \cs_gset_protected:Npn \hook_new_pair_with_args:nnn #1#2#3
227 {
228 }\EndIncludeInRelease

```

(End of definition for `\hook_new_pair:nn` and `\hook_new_pair_with_args:nnn`. These functions are documented on page [206](#).)

`\_\_hook\_include\_legacy\_code\_chunk:n` The L<sup>A</sup>T<sub>E</sub>X 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!

```

229 \IncludeInRelease{2023/06/01}{__hook_include_legacy_code_chunk:n}
230 {Hooks~with~args}
231 \cs_new_protected:Npn __hook_include_legacy_code_chunk:n #1
232 {

```

If the macro doesn't exist (which is the usual case) then nothing needs to be done.

```

233 \tl_if_exist:cT { @#1hook }
234 {

```

Of course if the legacy hook exists but is empty, there is no need to add anything under `legacy` the legacy label.

```

235 \tl_if_empty:cF { @#1hook }
236 {

```

Here we set `\_\_hook_replacing_args_false:` because no legacy code will reference hook arguments.

```

237 __hook_replacing_args_false:
238 \use:e
239 {
240 __hook_hook_gput_code_do:nnn {#1} { legacy }
241 { \exp_not:v { @#1hook } }
242 }
243 __hook_replacing_args_reset:

```

Once added to the hook, we need to clear it otherwise it might get added again later if the hook data gets updated.

```

244 __hook_tl_gclear:c { @#1hook }
245 }
246 }
247 }
248 \EndIncludeInRelease
249 \IncludeInRelease{2020/10/01}{__hook_include_legacy_code_chunk:n}

```

```

250 <latexrelease> {Hooks~with~args}
251 <latexrelease>\cs_gset_protected:Npn __hook_include_legacy_code_chunk:n #1
252 <latexrelease> {
253 <latexrelease> \tl_if_exist:cT { @#1hook }
254 <latexrelease> {
255 <latexrelease> \tl_if_empty:cF { @#1hook }
256 <latexrelease> {
257 <latexrelease> \exp_args:Nnnv __hook_hook_gput_code_do:nnn
258 <latexrelease> {#1} { legacy } { @#1hook }
259 <latexrelease> __hook_tl_gclear:c { @#1hook }
260 <latexrelease> }
261 <latexrelease> }
262 <latexrelease> }
263 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\__hook_include_legacy_code_chunk:n`.)

#### 4.4.4 Disabling and providing hooks

`\hook_disable_generic:n`  
`\__hook_disable:n`  
`\__hook_if_disabled_p:n`  
`\__hook_if_disabled:nTF`

Disables a hook by creating its `\g__hook_<hook>_declared_tl` so that the hook errors when used with `\hook_new:n`, then it undefines `\__hook_<hook>` so that it may not be executed.

This does not clear any code that may be already stored in the hook's structure, but doesn't allow adding more code. `\__hook_if_disabled:nTF` uses that specific combination to check if the hook is disabled.

```

264 <latexrelease>\IncludeInRelease{2021/06/01}{\hook_disable_generic:n}
265 <latexrelease> {Disable~hooks}
266 \cs_new_protected:Npn \hook_disable_generic:n #1
267 { __hook_normalize_hook_args:Nn __hook_disable:n {#1} }
268 \cs_new_protected:Npn __hook_disable:n #1
269 {
270 \tl_gclear_new:c { g__hook_#1_declared_tl }
271 \cs_undefine:c { __hook~#1 }
272 }
273 \prg_new_conditional:Npnn __hook_if_disabled:n #1 { p, T, F, TF }
274 {
275 \bool_lazy_and:nnTF
276 { \tl_if_exist_p:c { g__hook_#1_declared_tl } }
277 { ! \cs_if_exist_p:c { __hook~#1 } }
278 { \prg_return_true: }
279 { \prg_return_false: }
280 }
281 <latexrelease>\EndIncludeInRelease
282 <latexrelease>\IncludeInRelease{2020/10/01}{\hook_disable_generic:n}
283 <latexrelease> {Disable~hooks}
284 <latexrelease>
285 <latexrelease>\cs_new_protected:Npn \hook_disable_generic:n #1 {}
286 <latexrelease>
287 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\hook_disable_generic:n`, `\__hook_disable:n`, and `\__hook_if_disabled:nTF`.  
This function is documented on page 207.)

**\hook\_activate\_generic:n** The `\hook_activate_generic:n` declaration declares a new hook if it wasn't declared already, in which case it only checks that the already existing hook is not a reversed hook.

```

288 <|latexrelease>\IncludeInRelease{2023/06/01}{\hook_activate_generic:n}
289 <|latexrelease> {Providing~hooks}
290 \cs_new_protected:Npn \hook_activate_generic:n #1
291 { __hook_normalize_hook_args:Nn __hook_activate_generic:nn {#1} { } }
292 \cs_new_protected:Npn __hook_activate_generic:nn #1 #2
293 {

```

If the hook to be activated was disabled we warn (for now — this may change).

```

294 __hook_if_disabled:nTF {#1}
295 { \msg_warning:nnn { hooks } { activate-disabled } {#1} }

```

Otherwise we check if the hook is not declared, and if it isn't, figure out if it's reversed or not, then declare it accordingly.

```

296 {
297 __hook_if_declared:nF {#1}
298 {
299 \tl_new:c { g__hook_#1_declared_tl }
300 __hook_make_usable:nn {#1} { 0 }
301 \tl_gset:cx { g__hook_#1_reversed_tl }
302 { __hook_if_generic_reversed:nT {#1} { - } }

```

Reflect that we have activated the generic hook and set its execution code.

```

303 __hook_update_hook_code:n {#1}
304 }
305 }
306 }

307 <|latexrelease>\EndIncludeInRelease

308 <|latexrelease>\IncludeInRelease{2021/06/01}{\hook_activate_generic:n}
309 <|latexrelease> {Providing~hooks}
310 \cs_gset_protected:Npn __hook_activate_generic:nn #1 #2
311 <|latexrelease> {
312 <|latexrelease> __hook_if_disabled:nTF {#1}
313 <|latexrelease> { \msg_warning:nnn { hooks } { activate-disabled } {#1} }
314 <|latexrelease> {
315 <|latexrelease> __hook_if_declared:nF {#1}
316 <|latexrelease> {
317 <|latexrelease> \tl_new:c { g__hook_#1_declared_tl }
318 <|latexrelease> __hook_make_usable:n {#1}
319 <|latexrelease> \tl_gset:cx { g__hook_#1_reversed_tl }
320 <|latexrelease> { __hook_if_generic_reversed:nT {#1} { - } }
321 <|latexrelease> __hook_update_hook_code:n {#1}
322 <|latexrelease> }
323 <|latexrelease> }
324 <|latexrelease> }
325 <|latexrelease>\EndIncludeInRelease

326 <|latexrelease>\IncludeInRelease{2020/10/01}{\hook_activate_generic:n}
327 <|latexrelease> {Providing~hooks}
328 \cs_gset_protected:Npn \hook_activate_generic:n #1 { }
329 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\hook_activate_generic:n` and `\_hook_activate_generic:n`. This function is documented on page 207.)

## 4.5 Parsing a label

`\_hook_parse_label_default:nN`

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 by `\_hook_currname_or_default:`. #2 is a boolean representing if #1 is a label name.

```
330 \cs_new:Npn _hook_parse_label_default:nN #1#2
331 {
332 \tl_if_novalue:nTF {#1}
333 { _hook_currname_or_default: }
334 { \tl_trim_spaces_apply:nN {#1} _hook_parse_dot_label:nN #2 }
335 }
```

(End of definition for `\_hook_parse_label_default:nN`.)

`\_hook_parse_dot_label:nN`

`\_hook_parse_dot_label:w`

`\_hook_parse_dot_label_cleanup:w`

`\_hook_parse_dot_label_aux:w`

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:`. Otherwise the label is returned unchanged. #2 is a boolean representing if #1 is a label name.

```
336 \cs_new:Npn _hook_parse_dot_label:nN #1#2
337 {
338 \tl_if_empty:nTF {#1}
339 {
340 \bool_if:NTF #2
341 { \msg_expandable_error:nn { hooks } { empty-label } }
342 { \msg_expandable_error:nn { hooks } { empty-hook } }
343 _hook_currname_or_default:
344 }
345 {
346 \str_if_eq:nnTF {#1} { . }
347 { _hook_currname_or_default: }
348 { _hook_parse_dot_label:w #1 ./ \s__hook_mark }
349 }
350 }
351 \cs_new:Npn _hook_parse_dot_label:w #1 ./ #2 \s__hook_mark
352 {
353 \tl_if_empty:nTF {#1}
354 { _hook_parse_dot_label_aux:w #2 \s__hook_mark }
355 {
356 \tl_if_empty:nTF {#2}
357 { _hook_make_name:n {#1} }
358 { _hook_parse_dot_label_cleanup:w #1 ./ #2 \s__hook_mark }
359 }
360 }
361 \cs_new:Npn _hook_parse_dot_label_cleanup:w #1 ./ \s__hook_mark {#1}
362 \cs_new:Npn _hook_parse_dot_label_aux:w #1 ./ \s__hook_mark
363 { _hook_currname_or_default: / _hook_make_name:n {#1} }
```

(End of definition for `\_hook_parse_dot_label:nN` and others.)

`\__hook_currname_or_default:` This uses `\g__hook_hook_curr_name_tl` if it is set, otherwise it tries `\@currname`. If neither is set, it raises an error and uses the fallback value `label-missing`.

```

364 \cs_new:Npn __hook_currname_or_default:
365 {
366 \tl_if_empty:NTF \g__hook_hook_curr_name_tl
367 {
368 \tl_if_empty:NTF \@currname
369 {
370 \msg_expandable_error:nnn { latex2e } { should-not-happen }
371 {
372 \Empty~default-label.
373 __hook_make_name:n { label-missing }
374 }
375 }
376 { \g__hook_hook_curr_name_tl }
377 }

```

(End of definition for `\__hook_currname_or_default::`)

`\__hook_make_name:n` This provides a standard sanitization 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.

```

378 \cs_new:Npn __hook_make_name:n #1
379 {
380 \exp_after:wN \exp_after:wN \exp_after:wN __hook_make_name:w
381 \exp_after:wN \token_to_str:N \cs:w __hook~ #1 \cs_end:
382 }
383 \exp_last_unbraced:NNNNo
384 \cs_new:Npn __hook_make_name:w #1 \tl_to_str:n { __hook~ } { }

```

(End of 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`

This is the standard route for normalizing 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!`

```

385 \cs_new_protected:Npn __hook_normalize_hook_args_aux:Nn #1 #2
386 {
387 \group_begin:
388 \use:e
389 {
390 \group_end:
391 \exp_not:N #1 #2
392 }
393 }
394 \cs_new_protected:Npn __hook_normalize_hook_args:Nn #1 #2
395 {
396 __hook_normalize_hook_args_aux:Nn #1
397 { { __hook_parse_label_default:nN {#2} \c_false_bool } }
398 }

```

```

399 \cs_new_protected:Npn __hook_normalize_hook_args:Nnn #1 #2 #3
400 {
401 __hook_normalize_hook_args_aux:Nn #1
402 {
403 { __hook_parse_label_default:nN {#2} \c_false_bool }
404 { __hook_parse_label_default:nN {#3} \c_true_bool }
405 }
406 }
407 \cs_new_protected:Npn __hook_normalize_hook_rule_args:Nnnnn #1 #2 #3 #4 #5
408 {
409 __hook_normalize_hook_args_aux:Nn #1
410 {
411 { __hook_parse_label_default:nN {#2} \c_false_bool }
412 { __hook_parse_label_default:nN {#3} \c_true_bool }
413 { \tl_trim_spaces:n {#4} }
414 { __hook_parse_label_default:nN {#5} \c_true_bool }
415 }
416 }

```

(End of definition for `\__hook_normalize_hook_args:Nn` and others.)

`\__hook_curr_name_push:n`  
`\__hook_curr_name_push_aux:n`  
`\__hook_curr_name_pop:`  
`\__hook_end_document_label_check:`

The token list `\g__hook_hook_curr_name_tl` stores the name of the current package/file to be used as the default label in hooks. Providing a consistent interface is tricky because packages can be loaded within packages, and some packages may not use `\SetDefaultHookLabel` to change the default label (in which case `\@currname` is used).

To pull that one 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** (this stack should never go empty). If we're building the format, set the default label to be **top-level**:

```
417 \tl_gset:Nn \g__hook_hook_curr_name_tl { top-level }
```

Then, 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 first push the **top-level** entry:

```
418 ⟨latexrelease⟩ \seq_if_empty:NT \g__hook_name_stack_seq
419 ⟨latexrelease⟩ { \seq_gput_right:Nn \g__hook_name_stack_seq { top-level } }
```

then we dissect the `\@currnamestack`, adding `\@currname` to the stack:

```
420 ⟨latexrelease⟩ \cs_set_protected:Npn __hook_tmp:w #1 #2 #3
421 ⟨latexrelease⟩ {
422 ⟨latexrelease⟩ \quark_if_recursion_tail_stop:n {#1}
423 ⟨latexrelease⟩ \seq_gput_right:Nn \g__hook_name_stack_seq {#1}
424 ⟨latexrelease⟩ __hook_tmp:w
425 ⟨latexrelease⟩ }
426 ⟨latexrelease⟩ \exp_after:wN __hook_tmp:w \@currnamestack
427 ⟨latexrelease⟩ \q_recursion_tail \q_recursion_tail
428 ⟨latexrelease⟩ \q_recursion_tail \q_recursion_stop
```

and finally set the default label to be the `\@currname`:

```
429 ⟨latexrelease⟩ \tl_gset:Nx \g__hook_hook_curr_name_tl { \@currname }
430 ⟨latexrelease⟩ \seq_gpop_right:NN \g__hook_name_stack_seq \l__hook_tmpa_t1
```

Two commands keep track of the stack: when a file is input, `\__hook_curr_name_push:n` pushes the current default label onto the stack and sets the new default label (all in one go):

```

431 \cs_new_protected:Npn __hook_curr_name_push:n #1
432 { \exp_args:Nx __hook_curr_name_push_aux:n { __hook_make_name:n {#1} } }
433 \cs_new_protected:Npn __hook_curr_name_push_aux:n #1
434 {
435 \tl_if_blank:nTF {#1}
436 { \msg_error:nn { hooks } { no-default-label } }
437 {
438 \str_if_eq:nnTF {#1} { top-level }
439 {
440 \msg_error:nnnn { hooks } { set-top-level }
441 { to } { PushDefaultHookLabel } {#1}
442 }
443 {
444 \seq_gpush:NV \g__hook_name_stack_seq \g__hook_hook_curr_name_tl
445 \tl_gset:Nn \g__hook_hook_curr_name_tl {#1}
446 }
447 }
448 }

```

and when an input is over, the topmost item of the stack is popped, since that label will not be used again, and `\g__hook_hook_curr_name_tl` is updated to equal the now topmost item of the stack:

```

449 \cs_new_protected:Npn __hook_curr_name_pop:
450 {
451 \seq_gpop:NNTF \g__hook_name_stack_seq \l__hook_return_tl
452 { \tl_gset_eq:NN \g__hook_hook_curr_name_tl \l__hook_return_tl }
453 { \msg_error:nn { hooks } { extra-pop-label } }
454 }

```

At the end of the document we want to check if there was no `\__hook_curr_name_push:n` without a matching `\__hook_curr_name_pop:` (not a critical error, but it might indicate that something else is not quite right):

```

455 \tl_gput_right:Nn \@kernel@after@enddocument@afterlastpage
456 { __hook_end_document_label_check: }
457 \cs_new_protected:Npn __hook_end_document_label_check:
458 {
459 \seq_gpop:NNT \g__hook_name_stack_seq \l__hook_return_tl
460 {
461 \msg_error:nnx { hooks } { missing-pop-label }
462 { \g__hook_hook_curr_name_tl }
463 \tl_gset_eq:NN \g__hook_hook_curr_name_tl \l__hook_return_tl
464 __hook_end_document_label_check:
465 }
466 }

```

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.

```

467 \cs_new_protected:Npn __hook_set_default_hook_label:n #1
468 {
469 \seq_if_empty:NTF \g__hook_name_stack_seq
470 {
471 \msg_error:nnnn { hooks } { set-top-level }
472 { for } { SetDefaultHookLabel } {#1}

```

```

473 }
474 { \exp_args:Nx
475 __hook_set_default_label:n { __hook_make_name:n {#1} } }
476 }
477 \cs_new_protected:Npn __hook_set_default_label:n #1
478 {
479 \str_if_eq:nnTF {#1} { top-level }
480 {
481 \msg_error:nnnn { hooks } { set-top-level }
482 { to } { SetDefaultHookLabel } {#1}
483 }
484 { \tl_gset:Nn \g__hook_hook_curr_name_tl {#1} }
485 }

```

(End of definition for `\__hook_curr_name_push:n` and others.)

## 4.6 Adding or removing hook code

`\hook_gput_code:nnn`  
`\hook_gput_code_with_args:nnn`

```

__hook_gput_code:nnn
__hook_gput_code_store:nnn
__hook_gput_code:nnn
__hook_gput_code_do:nnn
__hook_prop_gput_labeled_cleanup:nnn
__hook_prop_gput_labeled_do:Nnnn
486 <|latexrelease> \IncludeInRelease{2023/06/01}{\hook_gput_code:nnn}
487 <|latexrelease> {Hooks~with~args}
488 \cs_new_protected:Npn \hook_gput_code:nnn #1 #2 #3
489 {
490 __hook_replacing_args_false:
491 __hook_normalize_hook_args:Nnn __hook_gput_code:nnn {#1} {#2} {#3}
492 __hook_replacing_args_reset:
493 }
494 \cs_new_protected:Npn \hook_gput_code_with_args:nnn #1 #2 #3
495 {
496 __hook_replacing_args_true:
497 __hook_normalize_hook_args:Nnn __hook_gput_code:nnn {#1} {#2} {#3}
498 __hook_replacing_args_reset:
499 }

```

If `\AddToHookWithArguments` was used, do some sanity checking, and if it's not possible to use arguments at this point, fall back to regular `\AddToHook` by using `\__hook_replacing_args_false`:

```

500 \cs_new_protected:Npn __hook_gput_code:nnn #1 #2 #3
501 {
502 __hook_chk_args_allowed:nn {#1} { AddToHook }

```

Then check if the code should be executed immediately, rather than stored:

```

503 __hook_if_execute_immediately:nTF {#1}
504 {

```

`\AddToHookWithArguments` can't be used on one-time hooks (that were already used).

```

505 __hook_if_replacing_args:TF
506 {
507 \msg_error:nnnn { hooks } { one-time-args }
508 {#1} { AddToHook }
509 }
510 { }
511 \use:n
512 }

```

```

513 { __hook_gput_code_store:nnn {#1} {#2} }
514 {#3}
515 }
516 \cs_new_protected:Npn __hook_gput_code_store:nnn #1 #2 #3
517 {

```

Then check if the hook is usable.

```
518 __hook_if_usable: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.

```

519 {
520 __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.

```

521 __hook_update_hook_code:n {#1}
522 }
```

If the hook is not usable, before giving up, check if it's not disabled and otherwise try to declare it as a generic hook, if its name matches one of the valid patterns.

```

523 {
524 __hook_if_disabled:nTF {#1}
525 { \msg_error:nnn { hooks } { hook-disabled } {#1} }
526 { __hook_try_declarng_generic_hook:nnn {#1} {#2} {#3} }
527 }
528 }
```

This macro will unconditionally add a chunk of code to the given hook.

```

529 \cs_new_protected:Npn __hook_hook_gput_code_do:nnn #1 #2 #3
530 {

```

However, first some debugging info if debugging is enabled:

```

531 __hook_debug:n{\iow_term:x{****~ Add~ to~
532 __hook_if_usable:nF {#1} { undeclared~ }
533 hook~ #1~ (#2)
534 \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. If the current label is `top-level`, the code is added to a dedicated token list `\__hook_toplevel_{hook}` that goes at the end of the hook (or at the beginning, for a reversed hook), just before `\__hook_next_{hook}`.

```

535 \str_if_eq:nnTF {#2} { top-level }
536 {
537 \str_if_eq:eeTF { top-level } { __hook_currname_or_default: }
538 {

```

If the hook's basic structure does not exist, we need to declare it with `\__hook_init_structure:n`.

```
539 __hook_init_structure:n {#1}
```

Then append to the `_toplevel` container for the hook.

```

540 __hook_cs_gput_right:nnn { _toplevel } {#1} {#3}
541 }
542 { \msg_error:nnn { hooks } { misused-top-level } {#1} }
543 }
544 {

```

When adding to the code pool, we have to double hashes if `\AddToHook` was used (`replacing_args` is false), so that later it is turned into a single parameter token, rather than a parameter to the hook macro.

```

545 \exp_args:Nx _hook_prop_gput_labeled_cleanup:nnn
546 {
547 _hook_if_replacing_args:TF
548 { \exp_not:n }
549 { _hook_double_hashes:n }
550 {#3}
551 }
552 {#1} {#2}
553 }
554 }

Adds code to a hook's code pool.

555 \cs_new_protected:Npn _hook_prop_gput_labeled_cleanup:nnn #1 #2 #3
556 {
557 \tl_set:Nn \l__hook_return_tl {#1}
558 _hook_if_replacing_args:TF
559 {
560 _hook_if_usable:nT {#2}
561 {
562 _hook_set_normalise_fn:nn {#2}
563 { Invalid~code~added~\msg_line_context: }
564 _hook_normalise_fn:nn {#3} {#1}
565 \prop_get:NnN \l__hook_work_prop {#3} \l__hook_return_tl
566 }
567 }
568 { }
569 \exp_args:NcV _hook_prop_gput_labeled_do:Nnn
570 { g__hook_#2_code_prop } \l__hook_return_tl {#3}
571 }
572 \cs_new_protected:Npn _hook_prop_gput_labeled_do:Nnn #1 #2 #3
573 {
574 \prop_get:NnNTF #1 {#3} \l__hook_return_tl
575 { \prop_gput:Nno #1 {#3} { \l__hook_return_tl #2 } }
576 { \prop_gput:Nnn #1 {#3} {#2} }
577 }

<|latexrelease> \EndIncludeInRelease

579 <|latexrelease> \IncludeInRelease{2020/10/01}{\hook_gput_code:nnn}
580 <|latexrelease> {Providing~hooks}
581 <|latexrelease> \cs_gset_protected:Npn \hook_gput_code:nnn #1 #2
582 <|latexrelease> { _hook_normalize_hook_args:Nnn
583 <|latexrelease> _hook_gput_code:nnn {#1} {#2} }
584 <|latexrelease> \cs_gset_protected:Npn _hook_gput_code:nnn #1 #2 #3
585 <|latexrelease> {
586 <|latexrelease> _hook_if_execute_immediately:nTF {#1}
587 <|latexrelease> {#3}
588 <|latexrelease> {
589 <|latexrelease> _hook_if_usable:nTF {#1}
590 <|latexrelease> {
591 <|latexrelease> _hook_hook_gput_code_do:nnn {#1} {#2} {#3}
592 <|latexrelease> _hook_update_hook_code:n {#1}

```

```

593 <|latexrelease> }
594 <|latexrelease> {
595 <|latexrelease> __hook_if_disabled:nTF {#1}
596 <|latexrelease> { \msg_error:nnn { hooks } { hook-disabled } {#1} }
597 <|latexrelease> { __hook_try_declaring_generic_hook:nnn
598 <|latexrelease> {#1} {#2} {#3} }
599 <|latexrelease> }
600 <|latexrelease> }
601 <|latexrelease> }
602 <|latexrelease>\cs_gset_protected:Npn __hook_hook_gput_code_do:nnn #1 #2 #3
603 <|latexrelease> {
604 <|latexrelease> __hook_debug:nf\iow_term:x{****~ Add~ to~
605 <|latexrelease> __hook_if_usable:nF {#1} { undeclared~ }
606 <|latexrelease> hook~ #1~ (#2)
607 <|latexrelease> \on@line\space <-- \tl_to_str:n{#3} }
608 <|latexrelease> \str_if_eq:nnTF {#2} { top-level }
609 <|latexrelease> {
610 <|latexrelease> \str_if_eq:eeTF { top-level }
611 <|latexrelease> { __hook_currname_or_default: }
612 <|latexrelease> {
613 <|latexrelease> __hook_init_structure:n {#1}
614 <|latexrelease> __hook_tl_gput_right:cn { __hook_toplevel~#1 } {#3}
615 <|latexrelease> }
616 <|latexrelease> { \msg_error:nnn { hooks } { misused-top-level } {#1} }
617 <|latexrelease> }
618 <|latexrelease> {
619 <|latexrelease> \prop_get:cnNTF
620 <|latexrelease> { g_hook_#1_code_prop } {#2} \l__hook_return_tl
621 <|latexrelease> {
622 <|latexrelease> \prop_gput:cno { g_hook_#1_code_prop } {#2}
623 <|latexrelease> { \l__hook_return_tl #3 }
624 <|latexrelease> }
625 <|latexrelease> { \prop_gput:cnn { g_hook_#1_code_prop } {#2} {#3} }
626 <|latexrelease> }
627 <|latexrelease> }
628 <|latexrelease>\cs_gset_protected:Npn \hook_gput_code_with_args:nnn #1#2#3 { }
629 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\hook_gput_code:nnn` and others. These functions are documented on page 208.)

`\_\_hook_chk_args_allowed:nn`

This macro checks if it is possible to add code with references to a hook's arguments for hook #1. It only does something if the function being run is `replacing_args`. This macro will error if the hook is declared and takes no arguments, then it will set `\_\_hook_replacing_args_false:` so that the macro which called it will add the code normally.

```

630 <|latexrelease>\IncludeInRelease{2023/06/01}{__hook_chk_args_allowed:nn}
631 <|latexrelease> {Hooks~with~args}
632 \cs_new_protected:Npn __hook_chk_args_allowed:nn #1 #2
633 {
634 __hook_if_replacing_args:TF
635 {
636 __hook_if_declared:nT {#1}
637 { \tl_if_empty:cT { c__hook_#1_parameter_tl } { \use_i:nn } }
638 \use_none:n

```

```

639 {
640 \msg_error:nnn { hooks } { without-args } {#1} {#2}
641 _hook_replacing_args_false:
642 }
643 }
644 {
645 }
646 \EndIncludeInRelease
647 \IncludeInRelease{2020/10/01}{_hook_chk_args_allowed:nn}
648 \{Hooks~with~args}
649 \cs_undefine:N _hook_chk_args_allowed:nn
650 \EndIncludeInRelease

```

(End of definition for \\_hook\_chk\_args\_allowed:nn.)

\\_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. An implicitly declared hook doesn't have arguments (in principle), so use \c\_false\_bool here.

```

651 \cs_new_protected:Npn _hook_gput_undeclared_hook:nnn #1 #2 #3
652 {
653 _hook_init_structure:n {#1}
654 _hook_gput_code_do:nnn {#1} {#2} {#3}
655 }

```

(End of definition for \\_hook\_gput\_undeclared\_hook:nnn.)

These entry-level macros just pass the arguments along to the common \\_hook\_try\_declaring\_generic\_hook:nNnn 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.

```

656 \EndIncludeInRelease{2023/06/01}
657 \IncludeInRelease{2023/06/01}{_hook_try_declaring_generic_hook:nnn}
658 \{Hooks~with~args}
659 \cs_new_protected:Npn _hook_try_declaring_generic_hook:nnn #1
660 {
661 _hook_try_declaring_generic_hook:wnTF #1 / / \scan_stop: {#1}
662 _hook_gput_code:nnn
663 _hook_gput_undeclared_hook:nnn
664 {#1}
665 }
666 \cs_new_protected:Npn _hook_try_declaring_generic_next_hook:nn #1
667 {
668 _hook_try_declaring_generic_hook:wnTF #1 / / \scan_stop: {#1}
669 _hook_gput_next_code:nn
670 _hook_gput_next_do:nn
671 {#1}

```

```

672 }
673 ⟨latexrelease⟩ \EndIncludeInRelease
674 ⟨latexrelease⟩ \IncludeInRelease{2021/11/15}
675 ⟨latexrelease⟩ {⟨_hook_try_declaring_generic_hook:nnn⟩}
676 ⟨latexrelease⟩ {Standardise-generic-hook-names}
677 ⟨latexrelease⟩ \cs_gset_protected:Npn ⟨_hook_try_declaring_generic_hook:nnn #1
678 ⟨latexrelease⟩ {
679 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook:wTF #1 / / / \scan_stop:
680 ⟨latexrelease⟩ {#1}
681 ⟨latexrelease⟩ ⟨hook_gput_code:nnn
682 ⟨latexrelease⟩ ⟨_hook_gput_undeclared_hook:nnn
683 ⟨latexrelease⟩ {#1}
684 ⟨latexrelease⟩ }
685 ⟨latexrelease⟩ \cs_gset_protected:Npn
686 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_next_hook:nn #1
687 ⟨latexrelease⟩ {
688 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook:wTF #1 / / / \scan_stop:
689 ⟨latexrelease⟩ {#1}
690 ⟨latexrelease⟩ ⟨hook_gput_next_code:nn
691 ⟨latexrelease⟩ ⟨_hook_gput_next_do:nn
692 ⟨latexrelease⟩ {#1}
693 ⟨latexrelease⟩ }
694 ⟨latexrelease⟩ \EndIncludeInRelease
695 ⟨latexrelease⟩ \IncludeInRelease{2020/10/01}
696 ⟨latexrelease⟩ {⟨_hook_try_declaring_generic_hook:nnn⟩}
697 ⟨latexrelease⟩ {Standardise-generic-hook-names}
698 ⟨latexrelease⟩ \cs_new_protected:Npn
699 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook:nnn #1
700 ⟨latexrelease⟩ {
701 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook:nNNnn {#1}
702 ⟨latexrelease⟩ ⟨hook_gput_code:nnn ⟨_hook_gput_undeclared_hook:nnn
703 ⟨latexrelease⟩ }
704 ⟨latexrelease⟩ \cs_new_protected:Npn
705 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_next_hook:nn #1
706 ⟨latexrelease⟩ {
707 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook:nNNnn {#1}
708 ⟨latexrelease⟩ ⟨hook_gput_next_code:nn ⟨_hook_gput_next_do:nn
709 ⟨latexrelease⟩ }

```

(End of definition for ⟨\_hook\_try\_declaring\_generic\_hook:nnn and  
 ⟨\_hook\_try\_declaring\_generic\_next\_hook:nn.)

⟨\_hook\_try\_declaring\_generic\_hook:nNNnn  
 hook\_try\_declaring\_generic\_hook\_split:nNNnn

⟨\_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.

```

710 ⟨latexrelease⟩ \cs_new_protected:Npn ⟨_hook_try_declaring_generic_hook:nNNnn #1
711 ⟨latexrelease⟩ {
712 ⟨latexrelease⟩ ⟨_hook_if_file_hook:wTF #1 / / \s_hook_mark
713 ⟨latexrelease⟩ {
714 ⟨latexrelease⟩ ⟨exp_args:Ne
715 ⟨latexrelease⟩ ⟨_hook_try_declaring_generic_hook_split:nNNnn
716 ⟨latexrelease⟩ { ⟨exp_args:Ne ⟨_hook_file_hook_normalize:n {#1} } }

```

```

717 <latexrelease> }
718 <latexrelease> { __hook_try_declarng_generic_hook_split:nNNnn {#1} }
719 <latexrelease> }

720 <latexrelease>\cs_new_protected:Npn
721 <latexrelease> __hook_try_declarng_generic_hook_split:nNNnn #1 #2 #3
722 <latexrelease> {
723 <latexrelease> __hook_try_declarng_generic_hook:wnTF #1 // \scan_stop:
724 <latexrelease> {#1}
725 <latexrelease> {#2}
726 <latexrelease> {#3} {#1}
727 <latexrelease> }
728 <latexrelease>\EndIncludeInRelease

(End of definition for __hook_try_declarng_generic_hook:nNNnn and
__hook_try_declarng_generic_hook_split:nNNnn.)
```

\\_\_hook\_try\_declarng\_generic\_hook:wnTF

```

729 <latexrelease>\IncludeInRelease{2023/06/01}
730 <latexrelease> {__hook_try_declarng_generic_hook:wn}
731 <latexrelease> {Hooks-with-args}
732 \prg_new_protected_conditional:Npnn
733 __hook_try_declarng_generic_hook:wn
734 #1 / #2 / #3 / #4 \scan_stop: #5 { TF }
735 {
736 __hook_if_generic:nTF {#5}
737 {
738 __hook_if_usable:nF {#5}
739 }
```

If the hook doesn't exist yet we check if it is a cmd hook and if so we attempt patching the command in addition to declaring the hook.

For some commands this will not be possible, in which case \\_\_hook\_patch\_cmd\_-or\_delay:Nnn (defined in ltcmdhooks) will generate an appropriate error message.

```

740 \str_if_eq:nnT {#1} { cmd }
741 {
742 __hook_try_put_cmd_hook:n {#5}
743 __hook_make_usable:nn {#5} { 9 }
744 \use_none:nnn
745 }
```

Declare the hook always even if it can't really be used (error message generated elsewhere).

Here we use \\_\_hook\_make\_usable:nn, so that a \hook\_new:n is still possible later. Generic hooks (except cmd hooks) take no arguments, so use zero as the second argument.

```

746 __hook_make_usable:nn {#5} { 0 }
747 }
748 __hook_if_generic_reversed:nT {#5}
749 { \tl_gset:cn { g_hook_#5_reversed_tl } { - } }
750 \prg_return_true:
751 }
752 }
```

Generic hooks are all named `<type>/<name>/<place>`, where `<type>` and `<place>` are predefined (`\c_hook_generic_<type>/./<place>_t1`), and `<name>` is the variable

component. Older releases had some hooks with the `<name>` in the third part, so the code below supports that syntax for a while, with a warning.

The `\exp_after:wN ... \exp:w` trick is there to remove the conditional structure inserted by `\_hook_try_declarng_generic_hook:wnTF` and thus allow access to the tokens that follow it, as is needed to keep things going.

When the deprecation cycle ends, the lines below should all be replaced by `\prg_return_false:`.

```

753 _hook_if_deprecated_generic:nTF {#5}
754 {
755 _hook_DEPRECATED_GENERIC_WARN:n {#5}
756 \exp_after:wN _hook_DECLARE_DEPRECATED_GENERIC:NNN
757 \exp:w % \exp_end:
758 }
759 { \prg_return_false: }
760 }
761 }
```

`\_hook_DEPRECATED_GENERIC_WARN:N` will issue a deprecation warning for a given hook, and mark that hook such that the warning will not be issued again (multiple warnings can be issued, but only once per hook).

```

762 \cs_new_protected:Npn _hook_DEPRECATED_GENERIC_WARN:n #1
763 { _hook_DEPRECATED_GENERIC_WARN:w #1 \s_hook_mark }
764 \cs_new_protected:Npn _hook_DEPRECATED_GENERIC_WARN:w
765 #1 / #2 / #3 \s_hook_mark
766 {
767 \ifcsexist:w __hook~#1/#2/#3 \cs_end: \else:
768 \msg_warning:nnnn { hooks } { generic-deprecated } {#1} {#2} {#3}
769 \fi:
770 \cs_gset_eq:cN { __hook~#1/#2/#3 } \scan_stop:
771 }
```

Now that the user has been told about the deprecation, we proceed by swapping `<name>` and `<place>` and adding the code to the correct hook.

```

772 \cs_new_protected:Npn _hook_DO_DEPRECATED_GENERIC:Nn #1 #2
773 { _hook_DO_DEPRECATED_GENERIC:Nw #1 #2 \s_hook_mark }
774 \cs_new_protected:Npn _hook_DO_DEPRECATED_GENERIC:Nw #1
775 #2 / #3 / #4 \s_hook_mark
776 { #1 { #2 / #4 / #3 } }
777 \cs_new_protected:Npn _hook_DECLARE_DEPRECATED_GENERIC:NNN #1 #2 #3
778 { _hook_DECLARE_DEPRECATED_GENERIC>NNW #1 #2 #3 \s_hook_mark }
779 \cs_new_protected:Npn _hook_DECLARE_DEPRECATED_GENERIC>NNW #1 #2
780 #3 / #4 / #5 \s_hook_mark
781 {
782 _hook_TRY_DECLARING_GENERIC_HOOK:WNTF #3 / #5 / #4 / \scan_stop:
783 { #3 / #5 / #4 }
784 #1 #2 { #3 / #5 / #4 }
785 }
786 \end{IncludeInRelease}

787 \EndIncludeInRelease{2021/11/15}
788 \begin{IncludeInRelease}{_hook_TRY_DECLARING_GENERIC_HOOK:WN}
789 \StandardiseGenericHookNames
790 \prg_new_protected_conditional:Nnn
```

```

791 <latexrelease> __hook_try_declarng_generic_hook:wn
792 <latexrelease> #1 / #2 / #3 / #4 \scan_stop: #5 { TF }
793 <latexrelease> {
794 <latexrelease> __hook_if_generic:nTF {#5}
795 <latexrelease> {
796 <latexrelease> __hook_if_usable:nF {#5}
797 <latexrelease> {
798 <latexrelease> \str_if_eq:nnT {#1} { cmd }
799 <latexrelease> { __hook_try_put_cmd_hook:n {#5} }
800 <latexrelease> __hook_make_usable:n {#5}
801 <latexrelease> }
802 <latexrelease> __hook_if_generic_reversed:nT {#5}
803 <latexrelease> { \tl_gset:cn { g__hook_#5_reversed_t1 } { - } }
804 <latexrelease> \prg_return_true:
805 <latexrelease>
806 <latexrelease>
807 <latexrelease> __hook_if_deprecated_generic:nTF {#5}
808 <latexrelease> {
809 <latexrelease> __hook_deprecated_generic_warn:n {#5}
810 <latexrelease> \exp_after:wN __hook_declare_deprecated_generic:NnN
811 <latexrelease> \exp:w % \exp_end:
812 <latexrelease>
813 <latexrelease> { \prg_return_false: }
814 <latexrelease>
815 <latexrelease> }
816 <latexrelease> \EndIncludeInRelease

817 <latexrelease> \IncludeInRelease{2021/06/01}
818 <latexrelease> __hook_try_declarng_generic_hook:wn
819 <latexrelease> {Support~cmd~hooks}
820 <latexrelease> \prg_new_protected_conditional:Npnn
821 <latexrelease> __hook_try_declarng_generic_hook:wn
822 <latexrelease> #1 / #2 / #3 / #4 \scan_stop: #5 { TF }
823 <latexrelease> {
824 <latexrelease> \tl_if_empty:nTF {#2}
825 <latexrelease> { \prg_return_false: }
826 <latexrelease> {
827 <latexrelease> \prop_if_in:NnTF \c__hook_genetics_prop {#1}
828 <latexrelease> {
829 <latexrelease> __hook_if_usable:nF {#5}
830 <latexrelease> {
831 <latexrelease> \str_if_eq:nnT {#1} { cmd }
832 <latexrelease> { __hook_try_put_cmd_hook:n {#5} }
833 <latexrelease> __hook_make_usable:n {#5}
834 <latexrelease>
835 <latexrelease> \prop_if_in:NnTF
836 <latexrelease> \c__hook_genetics_reversed_ii_prop {#2}
837 <latexrelease> { \tl_gset:cn { g__hook_#5_reversed_t1 } { - } }
838 <latexrelease>
839 <latexrelease> \prop_if_in:NnT
840 <latexrelease> \c__hook_genetics_reversed_iii_prop {#3}
841 <latexrelease> { \tl_gset:cn { g__hook_#5_reversed_t1 } { - } }
842 <latexrelease>
843 <latexrelease> \prg_return_true:
844 <latexrelease>

```

```

845 〈latexrelease〉 { \prg_return_false: }
846 〈latexrelease〉 }
847 〈latexrelease〉 }
848 〈latexrelease〉\EndIncludeInRelease

849 〈latexrelease〉\IncludeInRelease{2020/10/01}
850 〈latexrelease〉 {__hook_try_declar ing_generic_hook:wn}
851 〈latexrelease〉 {Support~cmd~hooks}
852 〈latexrelease〉\prg_new_protected_conditional:Npnn
853 〈latexrelease〉 __hook_try_declar ing_generic_hook:wn
854 〈latexrelease〉 #1 / #2 / #3 / #4 \scan_stop: #5 { TF }
855 〈latexrelease〉 {
856 〈latexrelease〉 \tl_if_empty:nTF {#2}
857 〈latexrelease〉 { \prg_return_false: }
858 〈latexrelease〉 {
859 〈latexrelease〉 \prop_if_in:NnTF \c__hook_genetics_prop {#1}
860 〈latexrelease〉 {
861 〈latexrelease〉 __hook_if_declared:nF {#5} { \hook_new:n {#5} }
862 〈latexrelease〉 \prop_if_in:NnTF
863 〈latexrelease〉 \c__hook_genetics_reversed_ii_prop {#2}
864 〈latexrelease〉 { \tl_gset:cn { g__hook_#5_reversed_tl } { - } }
865 〈latexrelease〉 {
866 〈latexrelease〉 \prop_if_in:NnT
867 〈latexrelease〉 \c__hook_genetics_reversed_iii_prop {#3}
868 〈latexrelease〉 { \tl_gset:cn { g__hook_#5_reversed_tl } { - } }
869 〈latexrelease〉 }
870 〈latexrelease〉 \prg_return_true:
871 〈latexrelease〉 }
872 〈latexrelease〉 { \prg_return_false: }
873 〈latexrelease〉 }
874 〈latexrelease〉 }
875 〈latexrelease〉\EndIncludeInRelease

```

(End of definition for `\_\_hook_try_declar ing_generic_hook:wnTF` and others.)

`\_\_hook_if_file_hook: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/foo.tex/before`). If it is a file-specific hook, then it executes the `true` branch, otherwise `false`.

```

876 〈latexrelease〉\IncludeInRelease{2021/11/15}{__hook_if_file_hook:w}
877 〈latexrelease〉 {Standardise-generic~hook~names}
878 〈latexrelease〉\EndIncludeInRelease
879 〈latexrelease〉\IncludeInRelease{2020/10/01}{__hook_if_file_hook:w}
880 〈latexrelease〉 {Standardise-generic~hook~names}
881 〈latexrelease〉\prg_new_conditional:Npnn __hook_if_file_hook:w
882 〈latexrelease〉 #1 / #2 / #3 \s_hook_mark { TF }
883 〈latexrelease〉 {
884 〈latexrelease〉 \str_if_eq:nnTF {#1} { file }
885 〈latexrelease〉 {
886 〈latexrelease〉 \bool_lazy_or:nnTF
887 〈latexrelease〉 { \tl_if_empty_p:n {#3} }
888 〈latexrelease〉 { \str_if_eq_p:nn {#3} { / } }
889 〈latexrelease〉 { \prg_return_false: }
890 〈latexrelease〉 {
891 〈latexrelease〉 \prop_if_in:NnTF \c__hook_genetics_file_prop {#2}
892 〈latexrelease〉 { \prg_return_true: }

```

```

893 <|latexrelease> { \prg_return_false: }
894 <|latexrelease> }
895 <|latexrelease> }
896 <|latexrelease> { \prg_return_false: }
897 <|latexrelease> }
898 <|latexrelease>\EndIncludeInRelease

(End of definition for __hook_if_file_hook:wTF.)

```

```

__hook_file_hook_normalize:n
__hook_strip_double_slash:n
__hook_strip_double_slash:w

```

```

899 <|latexrelease>\IncludeInRelease{2021/11/15}{__hook_file_hook_normalize:n}
900 <|latexrelease> {Standardise-generic-hook-names}
901 <|latexrelease>\EndIncludeInRelease

```

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./mypath//file.tex/after}`.

```

902 <|latexrelease>\IncludeInRelease{2020/10/01}{__hook_file_hook_normalize:n}
903 <|latexrelease> {Standardise-generic-hook-names}
904 <|latexrelease>\cs_new:Npn __hook_file_hook_normalize:n #1
905 <|latexrelease> { __hook_strip_double_slash:n {#1} }
906 <|latexrelease>\cs_new:Npn __hook_strip_double_slash:n #1
907 <|latexrelease> { __hook_strip_double_slash:w #1 // \s__hook_mark }

```

This function is always called after testing if the argument is a file hook with `\_\_hook_if_file_hook:wTF`, so we can assume it has three parts (it is either `file/.../before` or `file/.../after`), so we use `#1/#2/#3 //` instead of just `#1 //` to prevent losing a slash if the file name is empty.

```

908 <|latexrelease>\cs_new:Npn __hook_strip_double_slash:w #1/#2/#3//#4\s__hook_mark
909 <|latexrelease> {
910 <|latexrelease> \tl_if_empty:nTF {#4}
911 <|latexrelease> { #1/#2/#3 }
912 <|latexrelease> { __hook_strip_double_slash:w #1/#2/#3 /#4\s__hook_mark }
913 <|latexrelease> }
914 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\_\_hook_file_hook_normalize:n`, `\_\_hook_strip_double_slash:n`, and `\_\_hook_strip_double_slash:w`.)

```

\c__hook_generic_cmd/.before_tl
\c__hook_generic_cmd/.after_tl
\c__hook_generic_env/.before_tl
\c__hook_generic_env/.after_tl
\c__hook_generic_file/.before_tl
\c__hook_generic_file/.after_tl
\c__hook_generic_package/.before_tl
\c__hook_generic_package/.after_tl
\c__hook_generic_class/.before_tl
\c__hook_generic_class/.after_tl
\c__hook_generic_include/.before_tl
\c__hook_generic_include/.after_tl
\c__hook_generic_env/.begin_tl
\c__hook_generic_env/.end_tl
\c__hook_generic_include/.end_tl

```

Token lists defining the possible generic hooks. We don't provide any user interface to this as this is meant to be static.

**cmd** The generic hooks used for commands.

**env** The generic hooks used in `\begin` and `\end`.

**file, package, class, include** The generic hooks used when loading a file

```

915 <|latexrelease>\IncludeInRelease{2021/11/15}{\c__hook_generics_prop}
916 <|latexrelease> {Standardise-generic-hook-names}
917 \clist_map_inline:nn { cmd , env , file , package , class , include }
918 {
919 \tl_const:cn { c__hook_generic_#1/.before_tl } { + }
920 \tl_const:cn { c__hook_generic_#1/.after_tl } { - }

```

```

921 }
922 \tl_const:cn { c__hook_generic_env./begin_tl } { + }
923 \tl_const:cn { c__hook_generic_env./end_tl } { + }

924 \tl_const:cn { c__hook_generic_include./end_tl } { - }
925 \tl_const:cn { c__hook_generic_include./excluded_tl } { + }

Deprecated generic hooks:

926 \clist_map_inline:nn { file , package , class , include }
927 {
928 \tl_const:cn { c__hook_DEPRECATED_#1./before_tl } { }
929 \tl_const:cn { c__hook_DEPRECATED_#1./after_tl } { }
930 }
931 \tl_const:cn { c__hook_DEPRECATED_include./end_tl } { }
932 \EndIncludeInRelease

933 \IncludeInRelease{2020/10/01}{\c__hook_genetics_prop}
934 \Standardise-generic-hook-names
935 \prop_const_from_keyval:Nn \c__hook_genetics_prop
936 \cmd,env,file,package,class,include
937 \EndIncludeInRelease

```

(End of definition for `\c__hook_generic_cmd./before_tl` and others.)

The following generic hooks are supposed to use reverse ordering (the `ii` and `iii` names are kept for the deprecation cycle):

```

938 \IncludeInRelease{2021/11/15}{\c__hook_genetics_reversed_ii_prop}
939 \Standardise-generic-hook-names
940 \EndIncludeInRelease

941 \IncludeInRelease{2020/10/01}{\c__hook_genetics_reversed_ii_prop}
942 \Standardise-generic-hook-names
943 \prop_const_from_keyval:Nn
944 \c__hook_genetics_reversed_ii_prop {after=,end=}
945 \prop_const_from_keyval:Nn
946 \c__hook_genetics_reversed_iii_prop {after=}
947 \prop_const_from_keyval:Nn
948 \c__hook_genetics_file_prop {before=,after=}
949 \EndIncludeInRelease

```

(End of definition for `\c__hook_genetics_reversed_ii_prop`, `\c__hook_genetics_reversed_iii_prop`, and `\c__hook_genetics_file_prop`.)

Token lists defining the number of arguments for a given type of generic hook.

```

950 \IncludeInRelease{2023/06/01}{\c__hook_parameter_cmd./before_tl}
951 \Standardise-generic-hook-names

```

`cmd` hooks are declared with 9 arguments because they have a variable number of arguments (depending on the command they are attached to), so we use the maximum here.

```

952 \tl_const:cn { c__hook_parameter_cmd./before_tl } { #1#2#3#4#5#6#7#8#9 }
953 \tl_const:cn { c__hook_parameter_cmd./after_tl } { #1#2#3#4#5#6#7#8#9 }

954 \EndIncludeInRelease
955 \IncludeInRelease{2020/10/01}{\c__hook_parameter_cmd./before_tl}
956 \Standardise-generic-hook-names
957 \EndIncludeInRelease

```

(End of definition for \c\_\_hook\_parameter\_cmd./before\_t1 and  
\c\_\_hook\_parameter\_cmd./after\_t1.)

\hook\_gremove\_code:nn With \hook\_gremove\_code:nn{\langle hook\rangle}{\langle label\rangle} any code for \langle hook\rangle stored under \langle label\rangle is removed.

```
958 {\\@texrelease} \\IncludeInRelease{2023/06/01}{\\hook_gremove_code:nn}
959 {\\@texrelease} {\\@name{Hooks-with-args}}
960 \\cs_new_protected:Npn \\hook_gremove_code:nn #1 #2
961 { _hook_normalize_hook_args:Nnn _hook_gremove_code:nn {#1} {#2} }
962 \\cs_new_protected:Npn _hook_gremove_code:nn #1 #2
963 {
```

First check that the hook code pool exists. \\_hook\_if\_usable:nTF isn't used here because it should be possible to remove code from a hook before its defined (see section 2.1.8).

```
964 _hook_if_structure_exist:nTF {#1}
965 {
```

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}.

If all code is to be removed, clear the code pool \g\_\_hook\_{hook}\_code\_prop, the top-level code \\_hook\_toplevel\_{hook}, and the next-execution code \\_hook\_next\_{hook}.

```
966 \\str_if_eq:nnTF {#2} {*}
967 {
968 \\prop_gclear:c { g__hook_#1_code_prop }
969 _hook_toplevel_gset:nn {#1} { }
970 _hook_next_gset:nn {#1} { }
971 }
972 {
```

If the label is top-level then clear the token list, as all code there is under the same label.

```
973 \\str_if_eq:nnTF {#2} { top-level }
974 { _hook_toplevel_gset:nn {#1} { } }
975 {
976 \\prop_gpop:cnNF { g__hook_#1_code_prop }
977 {#2} \\l__hook_return_tl
978 { \\msg_warning:nnnn { hooks } { cannot-remove } {#1} {#2} }
979 }
980 }
```

Finally update the code, if the hook exists.

```
981 _hook_if_usable:nT {#1}
982 { _hook_update_hook_code:n {#1} }
983 }
```

If the code pool for this hook doesn't exist, show a warning:

```
984 {
985 _hook_if_deprecated_generic:nTF {#1}
986 {
987 _hook_deprecated_generic_warn:n {#1}
988 _hook_do_deprecated_generic:Nn
989 _hook_gremove_code:nn {#1} {#2}
990 }
```

```

991 { \msg_warning:nnnn { hooks } { cannot-remove } {#1} {#2} }
992 }
993 }
994 \EndIncludeInRelease
995 \IncludeInRelease{2020/10/01}{\hook_gremove_code:nn}
996 \textrm{\{Hooks-with-args\}}
997 \cs_new_protected:Npn _hook_gremove_code:nn #1 #2
998 \textrm{\{}
999 \textrm{\}}
1000 \textrm{\{}
1001 \textrm{\}}
1002 \textrm{\{}
1003 \textrm{\}}
1004 \textrm{\}}
1005 \textrm{\}}
1006 \textrm{\}}
1007 \textrm{\}}
1008 \textrm{\}}
1009 \textrm{\}}
1010 \textrm{\}}
1011 \textrm{\}}
1012 \textrm{\}}
1013 \textrm{\}}
1014 \textrm{\}}
1015 \textrm{\}}
1016 \textrm{\}}
1017 \textrm{\}}
1018 \textrm{\}}
1019 \textrm{\}}
1020 \textrm{\}}
1021 \textrm{\}}
1022 \textrm{\}}
1023 \textrm{\}}
1024 \textrm{\}}
1025 \textrm{\}}
1026 \textrm{\}}
1027 \textrm{\}}
1028 \textrm{\}}
1029 \textrm{\}}
1030 \textrm{\}}
1031 \EndIncludeInRelease

```

(End of definition for \hook\_gremove\_code:nn and \\_hook\_gremove\_code:nn. This function is documented on page 208.)

\\_hook\_cs\_gput\_right:nnn  
\\_\\_hook\\_cs\\_gput\\_right\\_fast:nnn  
\\_\\_hook\\_cs\\_gput\\_right\\_slow:nnn  
\\_\\_hook\\_code\\_gset\\_auxi:nnnn  
\\_\\_hook\\_code\\_gset\\_auxi:een

This macro is used to append code to the `toplevel` and `next` token lists, trating them correctly depending on their number of arguments, and depending if the code being added should have parameter tokens understood as parameters, or doubled to be stored as parameter tokens.

```

1032 \EndIncludeInRelease{2023/06/01}{_hook_cs_gput_right:nnn}
1033 \textrm{\{Hooks-with-args\}}

```

Check if the current hook is declared and takes no arguments. In this case, we short-circuit and use the simpler and much faster approach that doesn't require hash-doubling.

```

1034 \cs_new_protected:Npn __hook_cs_gput_right:nnn #1 #2
1035 {
1036 \if:w T
1037 __hook_if_declared:nF {#2} { F }
1038 \tl_if_empty:cF { c__hook_#2_parameter_tl } { F }
1039 T
1040 \exp_after:wN __hook_cs_gput_right_fast:nnn
1041 \else:
1042 \exp_after:wN __hook_cs_gput_right_slow:nnn
1043 \fi:
1044 {#1} {#2}
1045 }
1046 \cs_new_protected:Npn __hook_cs_gput_right_fast:nnn #1 #2 #3
1047 {
1048 \cs_gset:cp { __hook#1~#2 }
1049 { \exp_not:v { __hook#1~#2 } \exp_not:n {#3} } }
1050 \cs_new_protected:Npn __hook_cs_gput_right_slow:nnn #1 #2 #3
1051 {

```

The auxiliary `\__hook_code_gset_auxi:een` just does the assignment at the end. Its first argument is the parameter text of the macro, which is chosen here depending if `\c__hook_{hook}_parameter_tl` exists, if the hook is declared, and if it's a generic hook.

```

1051 \cs_if_exist:cF { __hook#1~#2 }
1052 { __hook_code_gset_aux:nnn {#1} {#2} { } }
1053 __hook_code_gset_auxi:een
1054 {
1055 __hook_if_declared:nTF {#2}
1056 { \tl_use:c { c__hook_#2_parameter_tl } }
1057 {
1058 __hook_if_generic:nTF {#2}
1059 { __hook_generic_parameter:n {#2} }
1060 { \c__hook_nine_parameters_tl }
1061 }
1062 }

```

Here we take the existing code in the macro, expand it with as many arguments as it takes, then double the hashes so the code can be reused.

PhO: Maybe can be improved.  
The case of adding to an empty cs  
can be optimised by quickly checking  
ing `\cs_replacement_spec`.

```

1063 {
1064 \exp_args:NNo \exp_args:No __hook_double_hashes:n
1065 {
1066 \cs:w __hook#1~#2 \exp_last_unbraced:Ne \cs_end:
1067 { __hook_braced_cs_parameter:n { __hook#1~#2 } }
1068 }
1069 }
```

Now the new code: if we are replacing arguments, then hashes are left untouched, otherwise they are doubled.

```

1070 {
1071 __hook_if_replacing_args:TF
1072 { \exp_not:n }
1073 { __hook_double_hashes:n }
1074 {#3}
1075 }
```

And finally, the csname which we'll define with all the above.

```

1076 { __hook#1~#2 }
1077 }
```

And as promised, the auxiliary that does the definition.

```

1078 \cs_new_protected:Npn __hook_code_gset_auxi:nnnn #1 #2 #3 #4
1079 { \cs_gset:cpn {#4} #1 { #2 #3 } }
1080 \cs_generate_variant:Nn __hook_code_gset_auxi:nnnn { een }

1081 \end{macro}
1082 \end{macro}
1083 \end{macro}
1084 \end{macro}
1085 \end{macro}
1086 \end{macro}
1087 \end{macro}
1088 \end{macro}

```

(End of definition for `\__hook_code_gput_right:nnn` and others.)

`\__hook_code_gset:nn`  
`\__hook_code_gset:ne`  
`\__hook_toplevel_gset:nn`  
`\__hook_next_gset:nn`  
`\__hook_code_gset_aux:nnn`

These macros define `\__hook(type)hook` (with `<type>` being `_next`, `_toplevel`, or empty) with the given code and the parameters stored in `\c__hook_<hook>_parameter_tl` (or none, if that doesn't exist).

```

1089 \end{macro}
1090 \end{macro}
1091 \end{macro}
1092 \end{macro}
1093 \end{macro}
1094 \end{macro}
1095 \end{macro}
1096 \end{macro}
1097 \end{macro}
1098 \end{macro}
1099 \end{macro}
1100 \end{macro}
1101 \end{macro}
1102 \end{macro}
1103 \end{macro}
1104 \end{macro}
1105 \end{macro}
1106 \end{macro}
1107 \end{macro}
1108 \end{macro}
1109 \end{macro}
1110 \end{macro}
1111 \end{macro}

```

(End of definition for `\__hook_code_gset:nn` and others.)

`\__hook_normalise_cs_args:nn`

This macro normalises the parameters of the macros `\__hook(type)hook` to take the right number of arguments after a hook is declared. At this point we know `\c__hook_<hook>_parameter_tl` exists, so use that to count the arguments and use that as `(parameter text)` for the newly (re)defined macro.

```

1112 \end{macro}
1113 \end{macro}
1114 \end{macro}
1115 \end{macro}
1116 \end{macro}

```

```

1117 {
1118 __hook_code_gset_auxi:een
1119 { \tl_use:c { c__hook#2_parameter_tl } }
1120 {
1121 \exp_args:NNo \exp_args:No __hook_double_hashes:n
1122 {
1123 \cs:w __hook#1~#2 \exp_last_unbraced:Ne \cs_end:
1124 { __hook_braced_cs_parameter:n { __hook#1~#2 } }
1125 }
1126 }
1127 {
1128 __hook#1~#2
1129 }
1130 }
1131 \langle latexrelease \rangle \EndIncludeInRelease
1132 \langle latexrelease \rangle \IncludeInRelease{2020/10/01}{__hook_normalise_cs_args:nn}
1133 \langle latexrelease \rangle \cs_new_protected:Npn __hook_normalise_cs_args:nn
1134 \langle latexrelease \rangle \cs_undefine:N __hook_normalise_cs_args:nn
1135 \langle latexrelease \rangle \EndIncludeInRelease

```

(End of definition for `\__hook_normalise_cs_args:nn`.)

`\__hook_normalise_code_pool:n  
\__hook_set_normalise_fn:nn`

This one's a bit of a hack. It takes a hook, and iterates over its code pool (`\g__hook_-<hook>_code_prop`), redefining each code label to use only valid arguments. This is used when, for example, a code is added referencing arguments #1 and #2, but the hook has only #1. In this example, every reference to #2 is changed to ##2. This is done because otherwise TeX will throw a low-level error every time some change happens to the hook (code is added, a rule is set, etc), which can get quite repetitive for no good reason.

```

1136 \langle latexrelease \rangle \IncludeInRelease{2023/06/01}{__hook_normalise_code_pool:n}
1137 \langle latexrelease \rangle \cs_new_protected:Npn __hook_normalise_code_pool:n #1
1138 \prop_clear:N \l__hook_work_prop
1139 {

```

First, call `\__hook_set_normalise_fn:nn` with the hook name to set everything up, then we'll loop over the hook's code pool applying the normalisation above. After that's done, copy the temporary property list back to the hook's.

```

1140 __hook_set_normalise_fn:nn {#1} { Offending-label:~'##1' }
1141 \prop_clear:N \l__hook_work_prop
1142 \prop_map_function:cN { g__hook#1_code_prop } __hook_normalise_fn:nn
1143 \prop_gset_eq:cN { g__hook#1_code_prop } \l__hook_work_prop
1144 }

```

The sole purpose of this function is to define `\__hook_normalise_fn:nn`, which will then do the correcting of the code being added to the hook.

```

1145 \cs_new_protected:Npn __hook_set_normalise_fn:nn #1 #2
1146 {

```

To start, we define two auxiliary token lists. `\l__hook_tmpb_tl` contains:

```

{\c__hook_hashes_tl 1}
{\c__hook_hashes_tl 2}
...
{\c__hook_hashes_tl 9}

```

```

1147 \cs_set:Npn __hook_tmp:w ##1##2##3##4##5##6##7##8##9 { }
1148 \tl_set:Ne \l__hook_tmpb_tl
1149 { __hook_braced_cs_parameter:n { __hook_tmp:w } }
1150 \group_begin:
1151 __hook_tl_set:cn { c__hook_hash_t1 } { \exp_not:N \c__hook_hashes_t1 }
1152 \use:e
1153 {
1154 \group_end:
1155 \tl_set:Nn \exp_not:N \l__hook_tmpb_tl { \l__hook_tmpb_tl }
1156 }
```

And `\l__hook_tmpa_t1` contains:

```

{\c__hook_hash_t1 1}
{\c__hook_hash_t1 2}
...
{\c__hook_hash_t1 <n>}
```

with `<n>` being the number of arguments declared for the hook.

```

1157 \exp_last_unbraced:Nf
1158 \cs_set:Npn __hook_tmp:w { __hook_parameter:n {#1} } { }
1159 \tl_set:Ne \l__hook_tmpa_tl
1160 { __hook_braced_cs_parameter:n { __hook_tmp:w } }
```

Now this function does the fun part. It is meant to be used with `\prop_map_function:NN`, taking a label name in `#1` and the code stored in that label in `#2`.

```

1161 \cs_gset_protected:Npx __hook_normalise_fn:nn ##1 ##2
1162 { }
```

Here we'll define two auxiliary macros: the first one throws an error when it detects an invalid argument reference. It piggybacks on TeX's low-level "Illegal parameter number" error, but it defines a weirdly-named control sequence so that the error comes out nicely formatted. For example, if the label "badpkg" adds some code that references argument `#3` in the hook "foo", which takes only two arguments, the error will be:

```

! Illegal parameter number in definition of hook 'foo'.
(hooks) Offending label: 'badpkg'.
<to be read again>
3
```

At the point of this definition, the error is raised if the code happens to reference an invalid argument. If it was possible to detect that this definition raised no error, the next step would be unnecessary. We'll do all this in a group so this weird definition doesn't leak out, and set `\tex_escapechar:D` to `-1` so this hack shows up extra nice in the case of an error.

```

1163 \group_begin:
1164 \int_set:Nn \tex_escapechar:D { -1 }
1165 \cs_set:cpn
1166 {
1167 hook~'#1'. ^^J
1168 (hooks) \prg_replicate:nn { 13 } { ~ }
1169 #2 % more message text
1170 }
1171 \exp_not:v { c__hook_#1_parameter_t1 }
1172 {##2}
1173 \group_end:
```

This next macro, with a much less fabulous name, takes always nine arguments, and it just transfers the code `##2` under the label `##1` to the temporary property list. The first `\n` arguments are taken from `\l_hook_tmpa_t1`, and the other  $9 - \n$  taken from `\l_hook_tmpb_t1` (which contains twice as many # tokens as the former). Then, `\_hook_double_hashes:n` is used to double non-argument hashes, and expand the `\c_hook_hash_t1` and `\c_hook_hashes_t1` to the actual parameter tokens.

```

1174 \cs_set:Npn \exp_not:N _hook_tmp:w
1175 \exp_not:V \c_hook_nine_parameters_t1
1176 {
1177 \prop_put:Nne \exp_not:N \l_hook_work_prop
1178 {##1} { \exp_not:N _hook_double_hashes:n {##2} }
1179 }

```

This next macro, with a much less fabulous name, takes always nine arguments, and it just transfers the code `##2` under the label `##1` to the temporary property list. The first `\n` arguments are taken from `\l_hook_tmpa_t1`, and the other  $9 - \n$  taken from `\l_hook_tmpb_t1` (which contains twice as many # tokens as the former). Then, `\_hook_double_hashes:n` is used to double non-argument hashes, and expand the `\c_hook_hash_t1` and `\c_hook_hashes_t1` to the actual parameter tokens.

```

1180 \exp_not:N _hook_tmp:w
1181 \exp_not:V \l_hook_tmpa_t1
1182 \exp_args:No \exp_not:o
1183 { \exp_after:wN _hook_tmp:w \l_hook_tmpb_t1 }
1184 }
1185 }
1186 \cs_new_eq:NN _hook_normalise_fn:nn ?
1187 \end{IncludeInRelease}
1188 \end{IncludeInRelease}{2020/10/01}{_hook_normalise_code_pool:n}
1189 \end{IncludeInRelease}{Hooks~with~args}
1190 \cs_undefine:N _hook_normalise_code_pool:n
1191 \end{IncludeInRelease}

```

Check if the expansion of a control sequence is empty by looking at its replacement text.

```

_hook_cs_if_empty_p:c
_hook_cs_if_empty:cTF
1192 \end{IncludeInRelease}{2023/06/01}{_hook_cs_if_empty:c}
1193 \end{IncludeInRelease}{Hooks~with~args}
1194 \prg_new_conditional:Npnn _hook_cs_if_empty:c #1 { p, T, F, TF }
1195 {
1196 \if:w \scan_stop: _hook_replacement_spec:c {#1} \scan_stop:
1197 \prg_return_true:
1198 \else:
1199 \prg_return_false:
1200 \fi:
1201 }
1202 \cs_new:Npn _hook_replacement_spec:c #1
1203 {
1204 \exp_args:Nc \token_if_macro:NT {#1}
1205 { \cs_replacement_spec:c {#1} }
1206 }
1207 \end{IncludeInRelease}
1208 \end{IncludeInRelease}{2020/10/01}{_hook_cs_if_empty:c}
1209 \end{IncludeInRelease}{Hooks~with~args}

```

```

1210 <latexrelease>\cs_undefine:N __hook_cs_if_empty:c
1211 <latexrelease>\EndIncludeInRelease

(End of definition for __hook_normalise_code_pool:n, __hook_set_normalise_fn:nn, and
__hook_cs_if_empty:cTF.)
```

\\_\_hook\_braced cs\_parameter:n  
\\_\_hook\_braced\_hidden\_loop:w  
\\_\_hook\_cs\_parameter\_count:N  
\\_\_hook\_cs\_parameter\_count:w  
\\_\_hook\_cs\_end:w

Looks at the *<parameter text>* of a control sequence, and returns a run of “hidden” braced parameters for that macro. This works as long as the macros take a simple run of zero to nine arguments. The parameters are “hidden” because the parameter tokens are returned inside \c\_\_hook\_hash\_t1 instead of explicitly, so that \\_\_hook\_double\_hashes:n won’t touch these.

```

1212 <latexrelease>\IncludeInRelease{2023/06/01}{__hook_braced_cs_parameter:n}
1213 <latexrelease> {Hooks~with~args}
1214 \cs_new:Npn __hook_braced_cs_parameter:n #1
1215 {
1216 \exp_last_unbraced:Ne __hook_braced_hidden_loop:w
1217 { \exp_args:Nc __hook_cs_parameter_count:N {#1} } ? \s__hook_mark
1218 }
1219 \cs_new:Npn __hook_braced_hidden_loop:w #1
1220 {
1221 \if:w ? #1
1222 __hook_use_i_delimit_by_s_mark:nw
1223 \fi:
1224 { \exp_not:N \c__hook_hash_t1 #1 }
1225 __hook_braced_hidden_loop:w
1226 }
1227 \cs_new:Npn __hook_cs_parameter_count:N #1
1228 {
1229 \exp_last_unbraced:Nf __hook_cs_parameter_count:w
1230 { \token_if_macro:NT #1 { \cs_parameter_spec:N #1 } }
1231 ? __hook_cs_end:w ? __hook_cs_end:w ? __hook_cs_end:w
1232 ? __hook_cs_end:w ? __hook_cs_end:w ? __hook_cs_end:w
1233 ? __hook_cs_end:w ? __hook_cs_end:w ? __hook_cs_end:w
1234 \s__hook_mark
1235 }
1236 \cs_new:Npn __hook_cs_parameter_count:w #1#2 #3#4 #5#6 #7#8
1237 { #2 #4 #6 #8 __hook_cs_parameter_count:w }
1238 \cs_new:Npn __hook_cs_end:w #1 \s__hook_mark { }
1239 <latexrelease>\EndIncludeInRelease
```

This function can’t be undefined when rolling back because it’s used at the end of this module to adequate the hook data structures to previous versions.

```

1240 <latexrelease>\IncludeInRelease{2020/10/01}{__hook_braced_cs_parameter:n}
1241 <latexrelease> {Hooks~with~args}
1242 <latexrelease>\EndIncludeInRelease

(End of definition for __hook_braced_cs_parameter:n and others.)
```

\\_\_hook\_braced\_parameter:n  
\\_\_hook\_braced\_real\_loop:w

This one is used in simpler cases, where no special handling of hashes is required. This is used only inside \\_\_hook\_initialize\_hook\_code:n, so it assumes \c\_\_hook\_<hook>\_parameter\_t1 is defined, but should work otherwise.

```

1243 <latexrelease>\IncludeInRelease{2023/06/01}{__hook_braced_parameter:n}
1244 <latexrelease> {Hooks~with~args}
1245 \cs_new:Npn __hook_braced_parameter:n #1
```

```

1246 {
1247 \if_case:w
1248 \int_eval:n
1249 { \exp_args:Nv \str_count:n { c__hook_#1_parameter_tl } / 3 }
1250 \exp_stop_f:
1251 \or: {##1}
1252 \or: {##1} {##2}
1253 \or: {##1} {##2} {##3}
1254 \or: {##1} {##2} {##3} {##4}
1255 \or: {##1} {##2} {##3} {##4} {##5}
1256 \or: {##1} {##2} {##3} {##4} {##5} {##6}
1257 \or: {##1} {##2} {##3} {##4} {##5} {##6} {##7}
1258 \or: {##1} {##2} {##3} {##4} {##5} {##6} {##7} {##8}
1259 \or: {##1} {##2} {##3} {##4} {##5} {##6} {##7} {##8} {##9}
1260 \else:
1261 \msg_expandable_error:nnn { latex2e } { should-not-happen }
1262 { Invalid-parameter-spec. }
1263 \fi:
1264 }
1265 \end{IncludeInRelease}
1266 \IncludeInRelease{2020/10/01}{__hook_braced_parameter:n}
1267 \end{IncludeInRelease} {Hooks-with-args}
1268 \cs_undefine:N __hook_braced_parameter:n
1269 \end{IncludeInRelease}

(End of definition for __hook_braced_parameter:n and __hook_braced_real_loop:w.)

```

\\_\\_hook\_parameter:n This is just a shortcut to e- or f-expand to the `<parameter text>` of the hook.

```

1270 \IncludeInRelease{2023/06/01}{__hook_parameter:n}
1271 \end{IncludeInRelease} {Hooks-with-args}
1272 \cs_new:Npn __hook_parameter:n #
1273 {
1274 \cs:w c__hook_
1275 \tl_if_exist:cTF { c__hook_#1_parameter_tl }
1276 { #1_parameter } { empty }
1277 _tl \cs_end:
1278 }
1279 \cs_new:Npn __hook_generic_parameter:n #
1280 { __hook_generic_parameter:w #1 / / \s__hook_mark }
1281 \cs_new:Npn __hook_generic_parameter:w #1 / #2 / #3 / #4 \s__hook_mark
1282 {
1283 \cs_if_exist_use:cF { c__hook_parameter_#1./.#3_tl }
1284 { \c__hook_empty_tl }
1285 }
1286 \end{IncludeInRelease}
1287 \IncludeInRelease{2020/10/01}{__hook_parameter:n}
1288 \end{IncludeInRelease} {Hooks-with-args}
1289 \cs_undefine:N __hook_parameter:n
1290 \cs_undefine:N __hook_generic_parameter:n
1291 \end{IncludeInRelease}

(End of definition for __hook_parameter:n.)

```

## 4.7 Setting rules for hooks code

```
\g__hook_??_code_prop
 __hook~??
\g__hook_??_reversed_tl
\c__hook_??_parameter_tl
```

Initially these variables simply used an empty “label” name (not two question marks). This was a bit unfortunate, because then 13doc 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, the code 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.

```
1292 \prop_new:c { g__hook_??_code_prop }
1293 \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.

```
1294 \tl_new:c { g__hook_??_reversed_tl }
```

The parameter text for the “default” hook is empty.

```
1295 (latexrelease)\IncludeInRelease{2023/06/01}{\c__hook_??_parameter_tl}
1296 (latexrelease) {Hooks~with~args}
1297 \tl_const:cn { c__hook_??_parameter_tl } { }
1298 (latexrelease)\EndIncludeInRelease
1299 (latexrelease)\IncludeInRelease{2020/10/01}{\c__hook_??_parameter_tl}
1300 (latexrelease) {Hooks~with~args}
1301 (latexrelease)\cs_undefine:c { c__hook_??_parameter_tl }
1302 (latexrelease)\EndIncludeInRelease
```

(End of definition for `\g__hook_??_code_prop` and others.)

`\hook_gset_rule:nnnn`  
`\__hook_gset_rule:nnnn`

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, which sets a default rule (to be used if no other relation between the two hooks exist).

```
1303 \cs_new_protected:Npn \hook_gset_rule:nnnn #1#2#3#4
1304 {
1305 __hook_normalize_hook_rule_args:Nnnnn __hook_gset_rule:nnnn
1306 {#1} {#2} {#3} {#4}
1307 }
1308 (latexrelease)\IncludeInRelease{2022/06/01}{__hook_gset_rule:nnnn}
1309 (latexrelease) {Refuse~setting~rule~for~one-time~hooks}
1310 \cs_new_protected:Npn __hook_gset_rule:nnnn #1#2#3#4
1311 {
1312 __hook_if_deprecated_generic:nT {#1}
1313 {
1314 __hook_DEPRECATED_GENERIC_WARN:n {#1}
1315 __hook_do_DEPRECATED_GENERIC:Nn __hook_gset_rule:nnnn {#1}
1316 {#2} {#3} {#4}
1317 __hook_use_NONE_DELIMIT_BY_S_MARK:w
1318 }
1319 __hook_if_EXECUTE_IMMEDIATELY:nT {#1}
1320 {
```

```

1321 \msg_error:nnnnn { hooks } { rule-too-late }
1322 {#1} {#2} {#3} {#4}
1323 _hook_use_none_delimit_by_s_mark:w
1324 }

```

First we ensure the basic data structure of the hook exists:

```
1325 _hook_init_structure:n {#1}
```

Then we clear any previous relationship between both labels.

```
1326 _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.

```

1327 \cs_if_exist_use:cTF { _hook_rule_#3_gset:nnn }
1328 {
1329 {#1} {#2} {#4}
1330 _hook_update_hook_code:n {#1}
1331 }
1332 {
1333 \msg_error:nnnnn { hooks } { unknown-rule }
1334 {#1} {#2} {#3} {#4}
1335 }
1336 \s__hook_mark
1337 }

1338 \end{IncludeInRelease}
1339 \begin{IncludeInRelease}[2020/10/01]{_hook_gset_rule:nnnn}
1340 \end{IncludeInRelease} [Refuse setting rule for one-time hooks]
1341 \cs_new_protected:Npn _hook_gset_rule:nnnn #1#2#3#4
1342 \begin{IncludeInRelease}
1343 _hook_if_deprecated_generic:nT {#1}
1344 \begin{IncludeInRelease}
1345 _hook_deprecated_generic_warn:n {#1}
1346 _hook_do_deprecated_generic:Nn _hook_gset_rule:nnnn
1347 {#1} {#2} {#3} {#4}
1348 \exp_after:wN \use:none:nnnnnnnn \use:none:n
1349 \end{IncludeInRelease}
1350 _hook_init_structure:n {#1}
1351 _hook_rule_gclear:nnn {#1} {#2} {#4}
1352 \cs_if_exist_use:cTF { _hook_rule_#3_gset:nnn }
1353 {
1354 {#1} {#2} {#4}
1355 _hook_update_hook_code:n {#1}
1356 }
1357 \begin{IncludeInRelease}
1358 \msg_error:nnnnn { hooks } { unknown-rule }
1359 {#1} {#2} {#3} {#4}
1360 \end{IncludeInRelease}
1361 \end{IncludeInRelease}
1362 \end{IncludeInRelease}

```

(End of definition for `\hook_gset_rule:nnnn` and `\_hook_gset_rule:nnnn`. This function is documented on page 209.)

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

```

_hook_rule_before_gset:nnn
_hook_rule_after_gset:nnn
_hook_rule_<_gset:nnn
_hook_rule_>_gset:nnn

```

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  $l_B \mid l_A$  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.

```

1363 \cs_new_protected:Npn __hook_rule_before_gset:nnn #1#2#3
1364 {
1365 __hook_tl_gset:cx
1366 { g_hook_#1_rule_ __hook_label_pair:nn {#2} {#3} _tl }
1367 { __hook_label_ordered:nnTF {#2} {#3} {<} {>} }
1368 }
1369 \cs_new_eq:cN { __hook_rule_<_gset:nnn } __hook_rule_before_gset:nnn
1370 \cs_new_protected:Npn __hook_rule_after_gset:nnn #1#2#3
1371 {
1372 __hook_tl_gset:cx
1373 { g_hook_#1_rule_ __hook_label_pair:nn {#3} {#2} _tl }
1374 { __hook_label_ordered:nnTF {#3} {#2} {<} {>} }
1375 }
1376 \cs_new_eq:cN { __hook_rule_>_gset:nnn } __hook_rule_after_gset:nnn

```

(End of 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.

```

1377 \cs_new_protected:Npn __hook_rule_voids_gset:nnn #1#2#3
1378 {
1379 __hook_tl_gset:cx
1380 { g_hook_#1_rule_ __hook_label_pair:nn {#2} {#3} _tl }
1381 { __hook_label_ordered:nnTF {#2} {#3} {>} {<} }
1382 }

```

(End of definition for `\_\_hook_rule_voids_gset:nnn`.)

`\_\_hook_rule_incompatible-error_gset:nnn` and `\_\_hook_rule_incompatible-warning_gset:nnn`

```

1383 \cs_new_protected:cpn { __hook_rule_incompatible-error_gset:nnn } #1#2#3
1384 { __hook_tl_gset:cn
1385 { g_hook_#1_rule_ __hook_label_pair:nn {#2} {#3} _tl }
1386 { xE }
1387 }
1388 \cs_new_protected:cpn { __hook_rule_incompatible-warning_gset:nnn } #1#2#3
1389 { __hook_tl_gset:cn
1390 { g_hook_#1_rule_ __hook_label_pair:nn {#2} {#3} _tl }
1391 { xW }
1392 }

```

(End of definition for `\_\_hook_rule_incompatible-error_gset:nnn` and `\_\_hook_rule_incompatible-warning_gset:nnn`.)

`\_\_hook_rule_unrelated_gset:nnn` and `\_\_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.

```

1393 \cs_new_protected:Npn __hook_rule_unrelated_gset:nnn #1#2#3 { }
1394 \cs_new_protected:Npn __hook_rule_gclear:nnn #1#2#3
1395 { \cs_undefine:c { g_hook_#1_rule_ __hook_label_pair:nn {#2} {#3} _tl } }

```

(End of definition for `\_hook_rule_unrelated_gset:nnn` and `\_hook_rule_gclear:nnn`.)

`\_hook_label_pair:nn` Ensure that the lexically greater label comes first.

```
1396 \cs_new:Npn _hook_label_pair:nn #1#2
1397 {
1398 \if_case:w _hook_str_compare:nn {#1} {#2} \exp_stop_f:
1399 #1 | #1 % 0
1400 \or: #1 | #2 % +1
1401 \else: #2 | #1 % -1
1402 \fi:
1403 }
```

(End of definition for `\_hook_label_pair:nn`.)

`\_hook_label_ordered_p:nn` 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.

```
1404 \prg_new_conditional:Npnn _hook_label_ordered:nn #1#2 { TF }
1405 {
1406 \if_int_compare:w _hook_str_compare:nn {#1} {#2} > 0 \exp_stop_f:
1407 \prg_return_true:
1408 \else:
1409 \prg_return_false:
1410 \fi:
1411 }
```

(End of definition for `\_hook_label_ordered:nnTF`.)

`\_hook_if_label_case:nnnn` To avoid doing the string comparison twice in `\_hook_initialize_single>NNn` (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.

```
1412 \cs_new:Npn _hook_if_label_case:nnnn #1#2
1413 {
1414 \cs:w use_
1415 \if_case:w _hook_str_compare:nn {#1} {#2}
1416 i \or: ii \else: iii \fi: :nnn
1417 \cs_end:
1418 }
```

(End of definition for `\_hook_if_label_case:nnnn`.)

`\_hook_update_hook_code:n` Before `\begin{document}` this does nothing, in the body it reinitializes the hook code using the altered data.

```
1419 \cs_new_eq:NN _hook_update_hook_code:n \use_none:n
```

(End of 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.

```
1420 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{_hook_initialize_all:}
1421 ⟨latexrelease⟩
1422 \cs_new_protected:Npn _hook_initialize_all:
1423 {
```

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.

```
1424 \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. Here we have to determine if the hook has arguments so that auxiliaries know what to do with hashes. We look at `\c__hook_<hook>_parameter_tl`, if it has any parameters, and set `replacing_args` accordingly.

```
1425 __hook_debug:n { \prop_gclear:N \g__hook_used_prop }
1426 \seq_map_inline:Nn \g__hook_all_seq
1427 {
1428 \tl_if_empty:cTF { \c__hook_##1_parameter_tl }
1429 { __hook_replacing_args_false: }
1430 { __hook_replacing_args_true: }
1431 __hook_update_hook_code:n {##1}
1432 __hook_replacing_args_reset:
1433 }
```

If we are debugging we show results hook by hook for all hooks that have data.

```
1434 __hook_debug:n
1435 {
1436 \iow_term:x { ^~J All~initialized~(non-empty)~hooks: }
1437 \prop_map_inline:Nn \g__hook_used_prop
1438 {
1439 \iow_term:x
1440 { ^~J ~ ##1 ~ -> ~ \cs_replacement_spec:c { __hook-##1 } ~ }
1441 }
1442 }
```

After all hooks are initialized we change the “use” to just call the hook code and not initialize it (as this was already done in the preamble).

```
1443 __hook_post_initialization_defs:
1444 }

1445 \end{macro}
1446 \EndIncludeInRelease
1447 \IncludeInRelease{2020/10/01}{__hook_initialize_all:
1448 {Hooks-with-args}}
1449 \cs_gset_protected:Npn __hook_initialize_all:
1450 __hook_initialize_all:
1451 __hook_update_hook_code:n
1452 __hook_initialize_hook_code:n
1453 __hook_debug:n { \prop_gclear:N \g__hook_used_prop }
1454 __hook_update_hook_code:n {##1}
1455 __hook_debug:n
1456 __hook_initialize_all:
1457 \iow_term:x{^~JAll~ initialized~ (non-empty)~ hooks:}
1458 \prop_map_inline:Nn \g__hook_used_prop
1459 __hook_update_hook_code:n
1460 __hook_initialize_hook_code:n
1461 __hook_debug:n
1462 __hook_update_hook_code:n {##1}
1463 __hook_initialize_all:
1464 }
```

```

1465 <|latexrelease> \cs_gset_eq:NN \hook_use:n __hook_use_initialized:n
1466 <|latexrelease> \cs_gset_eq:NN __hook_preamble_hook:n \use_none:n
1467 <|latexrelease> }
1468 <@=|
1469 <|latexrelease>\cs_gset_eq:NN \cexpl@@@initialize@all@0
1470 <|latexrelease> __hook_initialize_all:
1471 <@=hook>
1472 <|latexrelease>\EndIncludeInRelease

```

(End of 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.

```

1473 <|latexrelease>\IncludeInRelease{2023/06/01}{__hook_initialize_hook_code:n}
1474 <|latexrelease> {Hooks~with~args}
1475 \cs_new_protected:Npn __hook_initialize_hook_code:n #1
1476 {
1477 __hook_debug:n
1478 { \iow_term:x { ^~J Update~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.

```
1479 __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 top-level and next code chunks. If there are code chunks we call \\_\_hook\_initialize\_single:NNn 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.

```

1480 __hook_if_usable:nT {#1}
1481 {
1482 \prop_if_empty:cTF { g__hook_#1_code_prop }
1483 {
1484 __hook_code_gset:ne {#1}
1485 }

```

The hook may take arguments, so we add a run of braced parameters after the `_next` and `_toplevel` macros, so that the arguments passed to the hook are forwarded to them.

```

1486 \exp_not:c { __hook_toplevel~#1 }
1487 __hook_braced_parameter:n {#1}
1488 \exp_not:c { __hook_next~#1 }
1489 __hook_braced_parameter:n {#1}
1490 }
1491 }
1492 }
```

By default the algorithm sorts the code chunks and then saves the result in a token list for fast execution; this is done by adding the code chunks 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 that are used later on.

```
1493 __hook_if_reversed:nTF {#1}
```

```

1494 { \cs_set_eq:NN __hook_tl_gput:Nn __hook_tl_gput_left:Nn
1495 \cs_set_eq:NN __hook_clist_gput:NV \clist_gput_left:NV }
1496 { \cs_set_eq:NN __hook_tl_gput:Nn __hook_tl_gput_right:Nn
1497 \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 make a copy of the code property list that we can safely work on without changing the main one.

```

1498 \prop_set_eq:Nc \l_hook_work_prop { g_hook_#1_code_prop }
1499 __hook_initialize_single:ccn
1500 { __hook~#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.

```

1501 __hook_debug:n
1502 { \exp_args:NNx \prop_gput:Nnn \g_hook_used_prop {#1} { } }
1503 }
1504 }
1505 }
1506 \end{IncludeInRelease}
1507 \IncludeInRelease{2020/10/01}{__hook_initialize_hook_code:n}
1508 \begin{IncludeInRelease}
1509 \cs_gset_protected:Npn __hook_initialize_hook_code:n #1
1510 \begin{IncludeInRelease}
1511 __hook_debug:n
1512 { \iow_term:x { ^J Update~code~for~hook~'#1'
1513 \on@line :^J } }
1514 __hook_include_legacy_code_chunk:n {#1}
1515 __hook_if_usable:nT {#1}
1516 \begin{IncludeInRelease}
1517 \prop_if_empty:cTF { g_hook_#1_code_prop }
1518 {
1519 __hook_tl_gset:co { __hook~#1 }
1520 {
1521 \cs:w __hook_toplevel~#1 \exp_after:wN \cs_end:
1522 \cs:w __hook_next~#1 \cs_end:
1523 }
1524 }
1525 \begin{IncludeInRelease}
1526 __hook_if_reversed:nTF {#1}
1527 { \cs_set_eq:NN __hook_tl_gput:Nn
1528 __hook_tl_gput_left:Nn
1529 \cs_set_eq:NN __hook_clist_gput:NV
1530 \clist_gput_left:NV }
1531 { \cs_set_eq:NN __hook_tl_gput:Nn
1532 __hook_tl_gput_right:Nn
1533 \cs_set_eq:NN __hook_clist_gput:NV
1534 \clist_gput_right:NV }
1535 \prop_set_eq:Nc \l_hook_work_prop
1536 { g_hook_#1_code_prop }
1537 __hook_initialize_single:ccn
1538 { __hook~#1 } { g_hook_#1_labels_clist } {#1}

```

```

1539 ⟨latexrelease⟩ __hook_debug:n
1540 ⟨latexrelease⟩ { \exp_args:Nnx \prop_gput:Nnn \g__hook_used_prop
1541 ⟨latexrelease⟩ {#1} { } }
1542 ⟨latexrelease⟩ }
1543 ⟨latexrelease⟩ }
1544 ⟨latexrelease⟩ }
1545 ⟨latexrelease⟩\EndIncludeInRelease

(End of definition for __hook_initialize_hook_code:n.)

```

\\_\_hook\_tl\_cname:n It is faster to pass a single token and expand it when necessary than to pass a bunch of character tokens around.  
\\_\_hook\_seq\_cname:n

*FMi: note to myself: verify*

```
1546 \cs_new:Npn __hook_tl_cname:n #1 { __hook_label_#1_tl }
```

```
1547 \cs_new:Npn __hook_seq_cname:n #1 { __hook_label_#1_seq }
```

(End of definition for \\_\_hook\_tl\_cname:n and \\_\_hook\_seq\_cname: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:

```
1548 \seq_new:N \l__hook_labels_seq
```

- Number of labels used in the current hook. In Knuth's algorithm this is called  $N$ :

```
1549 \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  $R$ :

```
1550 \tl_new:N \l__hook_front_tl
1551 \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:

```
1552 \tl_new:c { __hook_tl_cname:n { 0 } }
```

(End of definition for \l\_\_hook\_labels\_seq and others.)

```
__hook_initialize_single:Nn
__hook_initialize_single:ccn
```

\\_\_hook\_initialize\_single:Nn implements the sorting of the code chunks for a hook and saves the result in the token list for fast execution (#4). The arguments are `<hook-code-plist>`, `<hook-code-tl>`, `<hook-top-level-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 ... maybe*

```
1553 <|latexrelease>\IncludeInRelease{2023/06/01}{_hook_initialize_single:Nn}
1554 <|latexrelease> {Hooks~with~args}
1555 \cs_new_protected:Npn _hook_initialize_single:Nn #1#2#3
1556 {
```

Step T1: Initialize the data structure ...

```
1557 \seq_clear:N \l__hook_labels_seq
1558 \int_zero:N \l__hook_labels_int
```

Store the name of the hook:

```
1559 \tl_set:Nn \l__hook_cur_hook_tl {\#3}
```

We loop over the property list holding the code and record all the labels listed there. Only the 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.

```
1560 \prop_map_inline:Nn \l__hook_work_prop
1561 {
1562 \int_incr:N \l__hook_labels_int
1563 \seq_put_right:Nn \l__hook_labels_seq {\##1}
1564 \l__hook_tl_set:cn { \l__hook_tl_cname:n {\##1} } { 0 }
1565 \seq_clear_new:c { \l__hook_seq_cname:n {\##1} }
1566 }
```

Steps T2 and T3: Here we 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 `\l__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 `\l__hook_apply_label_pair:nnn`, which takes the properly-ordered pair of labels as argument.

```
1567 \prop_map_inline:Nn \l__hook_work_prop
1568 {
1569 \prop_map_inline:Nn \l__hook_work_prop
1570 {
1571 \l__hook_if_label_case:nnnnn {\##1} {\####1}
1572 { \prop_map_break: }
1573 { \l__hook_apply_label_pair:nnn {\##1} {\####1} }
1574 { \l__hook_apply_label_pair:nnn {\####1} {\##1} }
1575 {\#3}
1576 }
1577 }
```

Now take a breath, and look at the data structures that have been set up:

```
1578 \l__hook_debug:n { \l__hook_debug_label_data:N \l__hook_work_prop }
```

Step T4:

```
1579 \tl_set:Nn \l__hook_rear_tl { 0 }
1580 \tl_set:cn { \l__hook_tl_cname:n { 0 } } { 0 }
1581 \seq_map_inline:Nn \l__hook_labels_seq
1582 {
```

```

1583 \int_compare:nNnT { \cs:w __hook_tl_cname:n {##1} \cs_end: } = 0
1584 {
1585 \tl_set:cn { __hook_tl_cname:n { \l__hook_rear_tl } }{##1}
1586 \tl_set:Nn \l__hook_rear_tl {##1}
1587 }
1588 }
1589 \tl_set_eq:Nc \l__hook_front_tl { __hook_tl_cname:n { 0 } }
1590 __hook_tl_gclear:N #1
1591 \clist_gclear:N #2

```

The whole loop gets combined in steps T5–T7:

```

1592 \bool_while_do:nn { ! \str_if_eq_p:Vn \l__hook_front_tl { 0 } }
1593 {

```

This part is step T5:

```

1594 \int_decr:N \l__hook_labels_int
1595 \prop_get:NVN \l__hook_work_prop \l__hook_front_tl \l__hook_return_tl
1596 \exp_args:NVN __hook_tl_gput:Nn #1 \l__hook_return_tl
1597 __hook_clist_gput:NV #2 \l__hook_front_tl
1598 __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.

```

1599 \seq_map_inline:cn { __hook_seq_cname:n { \l__hook_front_tl } }
1600 {
1601 \tl_set:cx { __hook_tl_cname:n {##1} }
1602 {
1603 \int_eval:n
1604 { \cs:w __hook_tl_cname:n {##1} \cs_end: - 1 }
1605 }
1606 \int_compare:nNnT
1607 { \cs:w __hook_tl_cname:n {##1} \cs_end: } = 0
1608 {
1609 \tl_set:cn
1610 { __hook_tl_cname:n { \l__hook_rear_tl } }{##1}
1611 \tl_set:Nn \l__hook_rear_tl {##1}
1612 }
1613 }

```

and here is step T7:

```

1613 \tl_set_eq:Nc \l__hook_front_tl
1614 { __hook_tl_cname: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.

```

1615 }
1616 \int_compare:nNnF \l__hook_labels_int = 0
1617 {
1618 \iow_term:x{=====}
1619 \iow_term:x{Error:~ label~ rules~ are~ incompatible:}

```

This is not really the information one needs in the error case but it will do for now

...

*FMi: improve output on a rainy day*

```

1620 __hook_debug_label_data:N \l__hook_work_prop
1621 \iow_term:x{=====}
1622 }

```

After we have added all hook code to #1, we finish it off by adding extra code for the **top-level** (#2) and for one time execution (#3). These should normally be empty. The **top-level** code is added with \\_\_hook\_tl\_gput:Nn as that might change for a reversed hook (then **top-level** is the very first code chunk added). The **next** code is always added last (to the right). The hook may take arguments, so we add a run of braced parameters after the **\_next** and **\_toplevel** macros, so that the arguments passed to the hook are forwarded to them.

```

1623 \exp_args:NNe __hook_tl_gput:Nn #1
1624 { \exp_not:c { __hook_toplevel~#3 } __hook_braced_parameter:n {#3} }
1625 __hook_tl_gput_right:Ne #1
1626 { \exp_not:c { __hook_next~#3 } __hook_braced_parameter:n {#3} }
1627 \use:e
1628 {
1629 \cs_gset:cpn { __hook~#3 } \use:c { c__hook_#3_parameter_tl }
1630 { \exp_not:V #1 }
1631 }
1632 }

1633 \cs_generate_variant:Nn __hook_initialize_single:NNn { cc }
1634 ⟨latexrelease⟩\EndIncludeInRelease
1635 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{__hook_initialize_single:NNn}
1636 ⟨latexrelease⟩
1637 ⟨latexrelease⟩\cs_new_protected:Npn __hook_initialize_single:NNn #1#2#3
1638 ⟨latexrelease⟩ {
1639 ⟨latexrelease⟩ \seq_clear:N \l__hook_labels_seq
1640 ⟨latexrelease⟩ \int_zero:N \l__hook_labels_int
1641 ⟨latexrelease⟩ \tl_set:Nn \l__hook_cur_hook_tl {#3}
1642 ⟨latexrelease⟩ \prop_map_inline:Nn \l__hook_work_prop
1643 ⟨latexrelease⟩ {
1644 ⟨latexrelease⟩ \int_incr:N \l__hook_labels_int
1645 ⟨latexrelease⟩ \seq_put_right:Nn \l__hook_labels_seq {##1}
1646 ⟨latexrelease⟩ __hook_tl_set:cn { __hook_tl_cname:n {##1} } { 0 }
1647 ⟨latexrelease⟩ \seq_clear_new:c { __hook_seq_cname:n {##1} }
1648 ⟨latexrelease⟩ }
1649 ⟨latexrelease⟩ \prop_map_inline:Nn \l__hook_work_prop
1650 ⟨latexrelease⟩ {
1651 ⟨latexrelease⟩ \prop_map_inline:Nn \l__hook_work_prop
1652 ⟨latexrelease⟩ {
1653 ⟨latexrelease⟩ __hook_if_label_case:nnnn {##1} {####1}
1654 ⟨latexrelease⟩ { \prop_map_break: }
1655 ⟨latexrelease⟩ { __hook_apply_label_pair:nnn {##1} {####1} }
1656 ⟨latexrelease⟩ { __hook_apply_label_pair:nnn {####1} {##1} }
1657 ⟨latexrelease⟩ {#3}
1658 ⟨latexrelease⟩ }
1659 ⟨latexrelease⟩
1660 ⟨latexrelease⟩ __hook_debug:n
1661 ⟨latexrelease⟩ { __hook_debug_label_data:N \l__hook_work_prop }
1662 ⟨latexrelease⟩ \tl_set:Nn \l__hook_rear_tl { 0 }
1663 ⟨latexrelease⟩ \tl_set:cn { __hook_tl_cname:n { 0 } } { 0 }
1664 ⟨latexrelease⟩ \seq_map_inline:Nn \l__hook_labels_seq

```

```

1665 <|latexrelease> {
1666 <|latexrelease> \int_compare:nNnT
1667 <|latexrelease> { \cs:w __hook_tl_cname:n {##1} \cs_end: } = 0
1668 <|latexrelease> {
1669 <|latexrelease> \tl_set:cn { __hook_tl_cname:n
1670 <|latexrelease> { \l__hook_rear_tl } } {##1}
1671 <|latexrelease> \tl_set:Nn \l__hook_rear_tl {##1}
1672 <|latexrelease> }
1673 <|latexrelease> }
1674 <|latexrelease> \tl_set_eq:Nc \l__hook_front_tl { __hook_tl_cname:n { 0 } }
1675 <|latexrelease> __hook_tl_gclear:N #1
1676 <|latexrelease> \clist_gclear:N #2
1677 <|latexrelease> \bool_while_do:nn
1678 <|latexrelease> { ! \str_if_eq_p:Vn \l__hook_front_tl { 0 } }
1679 <|latexrelease> {
1680 <|latexrelease> \int_decr:N \l__hook_labels_int
1681 <|latexrelease> \prop_get:NVN \l__hook_work_prop
1682 <|latexrelease> \l__hook_front_tl \l__hook_return_tl
1683 <|latexrelease> \exp_args:NNV __hook_tl_gput:Nn #1 \l__hook_return_tl
1684 <|latexrelease> __hook_clist_gput:NV #2 \l__hook_front_tl
1685 <|latexrelease> __hook_debug:n{ \iow_term:x
1686 <|latexrelease> {Handled~ code~ for~ \l__hook_front_tl} }
1687 <|latexrelease> \seq_map_inline:cn
1688 <|latexrelease> { __hook_seq_cname:n { \l__hook_front_tl } }
1689 <|latexrelease> {
1690 <|latexrelease> \tl_set:cx { __hook_tl_cname:n {##1} }
1691 <|latexrelease> { \int_eval:n
1692 <|latexrelease> { \cs:w __hook_tl_cname:n {##1} \cs_end: - 1 }
1693 <|latexrelease> }
1694 <|latexrelease> \int_compare:nNnT
1695 <|latexrelease> { \cs:w __hook_tl_cname:n {##1} \cs_end: } = 0
1696 <|latexrelease> {
1697 <|latexrelease> \tl_set:cn { __hook_tl_cname:n
1698 <|latexrelease> { \l__hook_rear_tl } } {##1}
1699 <|latexrelease> \tl_set:Nn \l__hook_rear_tl {##1}
1700 <|latexrelease> }
1701 <|latexrelease> }
1702 <|latexrelease> \tl_set_eq:Nc \l__hook_front_tl
1703 <|latexrelease> { __hook_tl_cname:n { \l__hook_front_tl } }
1704 <|latexrelease> }
1705 <|latexrelease> \int_compare:nNnF \l__hook_labels_int = 0
1706 <|latexrelease> {
1707 <|latexrelease> \iow_term:x{=====
1708 <|latexrelease> \iow_term:x{Error:~ label~ rules~ are~ incompatible:}
1709 <|latexrelease> __hook_debug_label_data:N \l__hook_work_prop
1710 <|latexrelease> \iow_term:x{=====}
1711 <|latexrelease> }
1712 <|latexrelease> \exp_args:NNo __hook_tl_gput:Nn #1
1713 <|latexrelease> { \cs:w __hook_toplevel~#3 \cs_end: }
1714 <|latexrelease> __hook_tl_gput_right:No #1 { \cs:w __hook_next~#3 \cs_end: }
1715 <|latexrelease> }
1716 <|latexrelease> \cs_generate_variant:Nn __hook_tl_gput_right:Nn { No }
1717 <|latexrelease> \EndIncludeInRelease

```

(End of definition for \\_\_hook\_initialize\_single:NNn.)

\\_\\_hook\\_tl\\_gput:Nn \\_\\_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.

```
1718 \cs_new:Npn __hook_tl_gput:Nn { \ERROR }
1719 \cs_new:Npn __hook_clist_gput:NV { \ERROR }
```

(End of definition for \\_\\_hook\\_tl\\_gput:Nn 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\\_\langle hook\rangle\\_rule\\_\#1\#\#2\\_t1, and not \g\\_\\_hook\\_\langle hook\rangle\\_rule\\_\#2\#\#1\\_t1. This also saves a great deal of time since we only need to check the order of the labels once.

The arguments here are \langle label1\rangle, \langle label2\rangle, \langle hook\rangle, and \langle hook-code-plist\rangle. 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 \langle hook\rangle, and if no rule is found, also try the \langle hook\rangle 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.

```
1720 \cs_new_protected:Npn __hook_apply_label_pair:nnn #1#2#3
1721 {
```

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.

```
1722 __hook_label_if_exist_apply:nnnF {#1} {#2} {#3}
1723 {
```

If there is no hook-specific rule we check for a default one and use that if it exists.

```
1724 __hook_label_if_exist_apply:nnnF {#1} {#2} { ?? } { }
1725 }
1726 }
1727 \cs_new_protected:Npn __hook_label_if_exist_apply:nnnF #1#2#3
1728 {
1729 \if_cs_exist:w g__hook_\#3_rule_\#1 | \#2_t1 \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.

```
1730 __hook_apply_rule:nnn {#1} {#2} {#3}
1731 \exp_after:wN \use_none:n
1732 \else:
1733 \use:nn
1734 \fi:
1735 }
```

(End of 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 \langle label1\rangle, \langle label2\rangle, \langle hook-name\rangle, and \langle hook-code-plist\rangle.

```
1736 \cs_new_protected:Npn __hook_apply_rule:nnn #1#2#3
1737 {
1738 \cs:w __hook_apply_
1739 \cs:w g__hook_\#3_reversed_t1 \cs_end: rule_
1740 \cs:w g__hook_\#3_rule_\#1 | \#2_t1 \cs_end: :nnn \cs_end:
```

```

1741 {#1} {#2} {#3}
1742 }

```

(End of definition for `\_\_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.

```

1743 \cs_new_protected:cpn { __hook_apply_rule_<:nnn } #1#2#3
1744 {
1745 __hook_debug:n { __hook_msg_pair_found:nnn {#1} {#2} {#3} }
1746 \tl_set:cx { __hook_tl_cname:n {#2} }
1747 { \int_eval:n{ \cs:w __hook_tl_cname:n {#2} \cs_end: + 1 } }
1748 \seq_put_right:cn{ __hook_seq_cname:n {#1} }{#2}
1749 }
1750 \cs_new_protected:cpn { __hook_apply_rule_>:nnn } #1#2#3
1751 {
1752 __hook_debug:n { __hook_msg_pair_found:nnn {#1} {#2} {#3} }
1753 \tl_set:cx { __hook_tl_cname:n {#1} }
1754 { \int_eval:n{ \cs:w __hook_tl_cname:n {#1} \cs_end: + 1 } }
1755 \seq_put_right:cn{ __hook_seq_cname:n {#2} }{#1}
1756 }

```

(End of definition for `\_\_hook\_apply\_rule_<:nnn` and `\_\_hook\_apply\_rule_>:nnn`.)

`\_\_hook\_apply\_rule_xE: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.

```

1757 \cs_new_protected:cpn { __hook_apply_rule_xE:nnn } #1#2#3
1758 {
1759 __hook_debug:n { __hook_msg_pair_found:nnn {#1} {#2} {#3} }
1760 \msg_error:nnnnnn { hooks } { labels-incompatible }
1761 {#1} {#2} {#3} { 1 }
1762 \use:c { __hook_apply_rule_->:nnn } {#1} {#2} {#3}
1763 \use:c { __hook_apply_rule_-<:nnn } {#1} {#2} {#3}
1764 }
1765 \cs_new_protected:cpn { __hook_apply_rule_xW:nnn } #1#2#3
1766 {
1767 __hook_debug:n { __hook_msg_pair_found:nnn {#1} {#2} {#3} }
1768 \msg_warning:nnnnnn { hooks } { labels-incompatible }
1769 {#1} {#2} {#3} { 0 }
1770 }

```

(End of definition for `\_\_hook\_apply\_rule_xE:nnn` and `\_\_hook\_apply\_rule_xW:nnn`.)

`\_\_hook\_apply\_rule_->:nnn` If we see  $\rightarrow$  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 put such empty code. However that would complicate the algorithm a lot with little gain.<sup>12</sup> 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 it instead from `\l__hook_cur_hook_tl` makes the default rules  $\rightarrow$  and  $\leftarrow$  work properly.

---

<sup>12</sup>This also has the advantage that the result of the sorting doesn't change, as it might otherwise do (for unrelated chunks) if we aren't careful.

```
1771 \cs_new_protected:cpn { __hook_apply_rule_->:nnn } #1#2#3
1772 {
1773 __hook_debug:n
1774 {
1775 __hook_msg_pair_found:nnn {#1} {#2} {#3}
1776 \iow_term:x{-->~ Drop~ '#2'~ code~ from~
1777 \iow_char:N \\ g__hook_ \l__hook_cur_hook_tl _code_prop ~
1778 because~ of~ '#1' }
1779 }
1780 \prop_put:Nnn \l__hook_work_prop {#2} { }
1781 }
1782 \cs_new_protected:cpn { __hook_apply_rule_-<:nnn } #1#2#3
1783 {
1784 __hook_debug:n
1785 {
1786 __hook_msg_pair_found:nnn {#1} {#2} {#3}
1787 \iow_term:x{-->~ Drop~ '#1'~ code~ from~
1788 \iow_char:N \\ g__hook_ \l__hook_cur_hook_tl _code_prop ~
1789 because~ of~ '#2' }
1790 }
1791 \prop_put:Nnn \l__hook_work_prop {#1} { }
1792 }
```

(End of definition for `\_hook\_apply\_rule\_->:nnn` and `\_hook\_apply\_rule_<:nnn`.)

Reversed rules.

```

__hook_apply_rule_>:nnn 1793 \cs_new_eq:cc { __hook_apply_rule_<:nnn } { __hook_apply_rule_>:nnn }
__hook_apply_rule_<:nnn 1794 \cs_new_eq:cc { __hook_apply_rule_>:nnn } { __hook_apply_rule_<:nnn }
__hook_apply_rule_>:nnn 1795 \cs_new_eq:cc { __hook_apply_rule_<:nnn } { __hook_apply_rule_<:nnn }
__hook_apply_rule_>:nnn 1796 \cs_new_eq:cc { __hook_apply_rule_>:nnn } { __hook_apply_rule_>:nnn }
__hook_apply_rule_xW:nnn 1797 \cs_new_eq:cc { __hook_apply_rule_xE:nnn } { __hook_apply_rule_xE:nnn }
__hook_apply_rule_xE:nnn 1798 \cs_new_eq:cc { hook apply rule xW:nnn } { hook apply rule xW:nnn }

```

(End of definition for `\_hook\_apply\_rule_<:nnn` and others.)

\\_\_hook\_msg\_pair\_found:nnn A macro to avoid moving this many tokens around.

```
1799 \cs_new_protected:Npn __hook_msg_pair_found:n #1#2#3
1800 {
1801 \iow_term:x{~ \str_if_eq:nnTF {#3} {??} {default} {~normal} ~
1802 rule~ __hook_label_pair:nn {#1} {#2}:~}
1803 \use:c { g__hook_#3_rule_ __hook_label_pair:nn {#1} {#2} _tl } ~
1804 found}
1805 }
```

(End of definition for \\_hook\\_msg\\_pair\\_found:nnn.)

\\_\\_hook\\_debug\\_label\\_data:N

```
1806 \cs_new_protected:Npn __hook_debug_label_data:N #1 {
1807 \iow_term:x{Code~ labels~ for~ sorting:}
1808 \iow_term:x{\~\seq_use:Nnnn\l__hook_labels_seq {~and~}{,}{~and~} }
1809 \iow_term:x{\~\J Data~ structure~ for~ label~ rules:}
1810 \prop_map_inline:Nn #1
1811 {
1812 \iow_term:x{\~##1~ =~ \tl_use:c{ __hook_tl_cname:n {##1} }~ ->~}
1813 \seq_use:cnnn{ __hook_seq_cname:n {##1} }{~->~}{~->~}{~->~}
```

```

1814 }
1815 }
1816 \iow_term:x{ }
1817 }

(End of definition for __hook_debug_label_data:N.)
```

**\hook\_show:n** 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.

**\\_\_hook\_log\_line:x**

```

1818 \cs_new_protected:Npn \hook_log:n #1
1819 {
1820 \cs_set_eq:NN __hook_log_cmd:x \iow_log:x
1821 __hook_normalize_hook_args:Nn __hook_log:nN {#1} \tl_log:x
1822 }
1823 \cs_new_protected:Npn \hook_show:n #1
1824 {
1825 \cs_set_eq:NN __hook_log_cmd:x \iow_term:x
1826 __hook_normalize_hook_args:Nn __hook_log:nN {#1} \tl_show:x
1827 }
1828 \cs_new_protected:Npn __hook_log_line:x #1
1829 { __hook_log_cmd:x { >#1 } }
1830 \cs_new_protected:Npn __hook_log_line_indent:x #1
1831 { __hook_log_cmd:x { >~\@spaces #1 } }

1832 (latexrelease)\IncludeInRelease{2023/06/01}{__hook_log:nN}
1833 (latexrelease) {Hooks~with~args}
1834 \cs_new_protected:Npn __hook_log:nN #1 #2
1835 {
1836 __hook_if_DEPRECATED_GENERIC:nT {#1}
1837 {
1838 __hook_DEPRECATED_GENERIC_WARN:n {#1}
1839 __hook_DO_DEPRECATED_GENERIC:Nn __hook_log:nN {#1} #2
1840 \exp_after:wN \use_none:nnnnnnnn \use_none:nnnn
1841 }
1842 __hook_preamble_hook:n {#1}
1843 __hook_log_cmd:x
1844 {
1845 ^~J ->~The~
1846 __hook_if_GENERIC:nT {#1} { generic~ }
1847 hook~'#1'
1848 __hook_if_DISABLED:nF {#1}
1849 {
1850 \exp_args:Nne __hook_print_args:nn {#1}
1851 {
1852 \int_eval:n
1853 { \str_count:e { __hook_PARAMETER:n {#1} } / 3 }
1854 }
1855 }
1856 :
1857 }

1858 __hook_if_USABLE:nF {#1}
1859 { __hook_log_line:x { The~hook~is~not~declared. } }
1860 __hook_if_DISABLED:nT {#1}
```

```

1861 { __hook_log_line:x { The~hook~is~disabled. } }
1862 \hook_if_empty:nTF {#1}
1863 { #2 { The~hook~is~empty } }
1864 {
1865 __hook_log_line:x { Code~chunks: }
1866 \bool_lazy_or:nnTF
1867 { ! \prop_if_exist_p:c { g__hook_#1_code_prop } }
1868 { \prop_if_empty_p:c { g__hook_#1_code_prop } }
1869 { __hook_log_line_indent:x { --- } }
1870 {
1871 \prop_map_inline:cn { g__hook_#1_code_prop }
1872 {
1873 \exp_after:wN \cs_set:Npn \exp_after:wN __hook_tmp:w
1874 \c__hook_nine_parameters_tl {##2}
1875 __hook_log_line_indent:x
1876 { ##1~->~\cs_replacement_spec:N __hook_tmp:w }
1877 }
1878 }

```

If there is code in the top-level token list, print it:

```

1879 __hook_log_line:x
1880 {
1881 Document-level~(top-level)~code
1882 __hook_if_usable:nT {#1}
1883 { ~(executed~__hook_if_reversed:nTF {#1} {first} {last}) :
1884 }
1885 __hook_log_line_indent:x
1886 {
1887 __hook_cs_if_empty:cTF { __hook_toplevel~#1 }
1888 { --- }
1889 { -> ~ \cs_replacement_spec:c { __hook_toplevel~#1 } }
1890 }
1891 __hook_log_line:x { Extra~code~for~next~invocation: }
1892 __hook_log_line_indent:x
1893 {
1894 __hook_cs_if_empty:cTF { __hook_next~#1 }
1895 { --- }

```

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.

```

1896 {
1897 -> ~ \exp_last_unbraced:Nf __hook_log_next_code:w
1898 { \cs_replacement_spec:c { __hook_next~#1 } }
1899 }
1900

```

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.

```

1901 __hook_log_line:x { Rules: }
1902 \bool_set_true:N \l__hook_tmpa_bool
1903 __hook_list_rules:nn {#1}
1904 {
1905 \bool_set_false:N \l__hook_tmpa_bool

```

```

1906 __hook_log_line_indent:x
1907 {
1908 ##2~ with~
1909 \str_if_eq:nNT {##3} {??} { default~ }
1910 relation~ ##1
1911 }
1912 }
1913 \bool_if:NT \l__hook_tmpa_bool
1914 { __hook_log_line_indent:x { --- } }

```

When the hook is declared (that is, the sorting algorithm is applied to that hook) and not empty

```

1915 \bool_lazy_and:nnTF
1916 { __hook_if_usable_p:n {#1} }
1917 { ! \hook_if_empty_p:n {#1} }
1918 {
1919 __hook_log_line:x
1920 {
1921 Execution~order
1922 \bool_if:NTF \l__hook_tmpa_bool
1923 { __hook_if_reversed:nT {#1} { ~(after~reversal) } }
1924 { ~(after~
1925 __hook_if_reversed:nT {#1} { reversal~and~ }
1926 applying~rules)
1927 } :
1928 }
1929 #2 % \tl_show:n
1930 {
1931 \@spaces
1932 \clist_if_empty:cTF { g__hook_#1_labels_clist }
1933 { --- }
1934 { \clist_use:cn { g__hook_#1_labels_clist } { ,~ } }
1935 }
1936 }
1937 {
1938 __hook_log_line:x { Execution~order: }
1939 #2
1940 {
1941 \@spaces Not~set~because~the~hook~ __hook_if_usable:nTF {#1}
1942 { code~pool~is~empty }
1943 { is~__hook_if_disabled:nTF {#1} {disabled} {undeclared} }
1944 }
1945 }
1946 }
1947 }
1948 <|latexrelease> \EndIncludeInRelease
1949 %
1950 <|latexrelease> \IncludeInRelease{2020/10/01}{__hook_log:nN}
1951 <|latexrelease> {Hooks~with~args}
1952 <|latexrelease> \cs_new_protected:Npn __hook_log:nN #1 #2
1953 <|latexrelease> {
1954 <|latexrelease> __hook_if_deprecated_generic:NT {#1}
1955 <|latexrelease> {
1956 <|latexrelease> __hook_DEPRECATED_GENERIC_WARN:n {#1}

```

```

1957 〈latexrelease〉 __hook_do_deprecated_generic:Nn __hook_log:nN {#1} #2
1958 〈latexrelease〉 \exp_after:wN \use_none:nnnnnnnn \use_none:nnnn
1959 〈latexrelease〉 }
1960 〈latexrelease〉 __hook_preamble_hook:n {#1}
1961 〈latexrelease〉 __hook_log_cmd:x
1962 〈latexrelease〉 { ^^J ->~The~ __hook_if_generic:nT
1963 〈latexrelease〉 {#1} { generic~ } hook~'#1': }
1964 〈latexrelease〉 __hook_if_usable:nF {#1}
1965 〈latexrelease〉 { __hook_log_line:x { The~hook~is~not~declared. } }
1966 〈latexrelease〉 __hook_if_disabled:nT {#1}
1967 〈latexrelease〉 { __hook_log_line:x { The~hook~is~disabled. } }
1968 〈latexrelease〉 \hook_if_empty:nTF {#1}
1969 〈latexrelease〉 { #2 { The~hook~is~empty } }
1970 〈latexrelease〉 {
1971 〈latexrelease〉 __hook_log_line:x { Code~chunks: }
1972 〈latexrelease〉 \prop_if_empty:cTF { g__hook_#1_code_prop }
1973 〈latexrelease〉 { __hook_log_indent:x { --- } }
1974 〈latexrelease〉 {
1975 〈latexrelease〉 \prop_map_inline:cn { g__hook_#1_code_prop }
1976 〈latexrelease〉 { __hook_log_line_indent:x
1977 〈latexrelease〉 { ##1~->~\tl_to_str:n {##2} } }
1978 〈latexrelease〉 }
1979 〈latexrelease〉 __hook_log_line:x
1980 〈latexrelease〉 {
1981 〈latexrelease〉 Document-level~(top-level)~code
1982 〈latexrelease〉 __hook_if_usable:nT {#1}
1983 〈latexrelease〉 { ~(executed~
1984 〈latexrelease〉 __hook_if_reversed:nTF {#1} {first} {last}) } :
1985 〈latexrelease〉 }
1986 〈latexrelease〉 __hook_log_line_indent:x
1987 〈latexrelease〉 {
1988 〈latexrelease〉 \tl_if_empty:cTF { __hook_toplevel~#1 }
1989 〈latexrelease〉 { --- }
1990 〈latexrelease〉 { -> ~ \exp_args:Nv \tl_to_str:n
1991 〈latexrelease〉 { __hook_toplevel~#1 } }
1992 〈latexrelease〉 }
1993 〈latexrelease〉 __hook_log_line:x { Extra~code~for~next~invocation: }
1994 〈latexrelease〉 __hook_log_line_indent:x
1995 〈latexrelease〉 {
1996 〈latexrelease〉 \tl_if_empty:cTF { __hook_next~#1 }
1997 〈latexrelease〉 { --- }
1998 〈latexrelease〉 { -> ~ \exp_args:Nv __hook_log_next_code:n
1999 〈latexrelease〉 { __hook_next~#1 } }
2000 〈latexrelease〉 }
2001 〈latexrelease〉 __hook_log_line:x { Rules: }
2002 〈latexrelease〉 \bool_set_true:N \l__hook_tmpa_bool
2003 〈latexrelease〉 __hook_list_rules:nn {#1}
2004 〈latexrelease〉 {
2005 〈latexrelease〉 \bool_set_false:N \l__hook_tmpa_bool
2006 〈latexrelease〉 __hook_log_line_indent:x
2007 〈latexrelease〉 {
2008 〈latexrelease〉 ##2~ with~
2009 〈latexrelease〉 \str_if_eq:nnT {##3} {??} { default~ }
2010 〈latexrelease〉 relation~ ##1

```

```

2011 <|latexrelease> }
2012 <|latexrelease> }
2013 <|latexrelease> \bool_if:NT \l__hook_tmpa_bool
2014 <|latexrelease> { __hook_log_line_indent:x { --- } }
2015 <|latexrelease> \bool_lazy_and:nnTF
2016 <|latexrelease> { __hook_if_usable_p:n {#1} }
2017 <|latexrelease> { ! \hook_if_empty_p:n {#1} }
2018 <|latexrelease> {
2019 <|latexrelease> __hook_log_line:x
2020 <|latexrelease> {
2021 <|latexrelease> __Execution-order
2022 <|latexrelease> \bool_if:NTF \l__hook_tmpa_bool
2023 <|latexrelease> { __hook_if_reversed:nT
2024 <|latexrelease> { #1}{ ~ (after-reversal) } }
2025 <|latexrelease> { ~ (after-
2026 <|latexrelease> __hook_if_reversed:nT {#1} { reversal-and-
2027 <|latexrelease> applying-rules)
2028 <|latexrelease> } :
2029 <|latexrelease> }
2030 <|latexrelease> #2 % \tl_show:n
2031 <|latexrelease> {
2032 <|latexrelease> \@spaces
2033 <|latexrelease> \clist_if_empty:cTF { g__hook_#1_labels_clist }
2034 <|latexrelease> { --- }
2035 <|latexrelease> { \clist_use:cn
2036 <|latexrelease> { g__hook_#1_labels_clist } { ,~ } }
2037 <|latexrelease> }
2038 <|latexrelease> }
2039 <|latexrelease> {
2040 <|latexrelease> __hook_log_line:x { __Execution-order: }
2041 <|latexrelease> #2
2042 <|latexrelease> {
2043 <|latexrelease> \@spaces Not-set-because-the-hook-
2044 <|latexrelease> __hook_if_usable:nTF {#1}
2045 <|latexrelease> { code-pool-is-empty }
2046 <|latexrelease> { is-__hook_if_disabled:nTF
2047 <|latexrelease> { #1} {disabled} {undeclared} }
2048 <|latexrelease> }
2049 <|latexrelease> }
2050 <|latexrelease> }
2051 <|latexrelease> }
2052 <|latexrelease> \EndIncludeInRelease

```

To display the code for next invocation only (i.e., from \AddToHookNext we have to remove the string \\_\_hook\_clear\_next:n{*hook*}, so the simplest is to use a macro delimited by a }<sub>12</sub>.

```

2053 <|latexrelease> \IncludeInRelease{2023/06/01}{__hook_log_next_code:n}
2054 <|latexrelease> {Hooks-with-args}
__hook_log_next_code:n 2055 \exp_last_unbraced:NNNN
2056 \cs_new:Npn __hook_log_next_code:w #1 \c_right_brace_str { }
2057 <|latexrelease> \EndIncludeInRelease
2058 <|latexrelease> \IncludeInRelease{2020/10/01}{__hook_log_next_code:n}
2059 <|latexrelease> {Hooks-with-args}
2060 <|latexrelease> \cs_gset:Npn __hook_log_next_code:n #1

```

```

2061 <latexrelease> { \exp_args:No \tl_to_str:n { \use_none:nn #1 } }
2062 <latexrelease>\EndIncludeInRelease

```

Pretty-prints the number of arguments of a hook.

```

2063 \cs_new:Npn __hook_print_args:nn #1 #2
2064 {
2065 \int_compare:nNnT {#2} > { 0 }
2066 {
2067 __hook_if_declared:nT {#1} { \use_none:nnn }
2068 __hook_if_cmd_hook:nT {#1}
2069 { \use_i:nnn { ~ (unknown ~) }
2070 \use:n { ~ (#2 ~) }
2071 argument \int_compare:nNnT {#2} > { 1 } { s })
2072 }
2073 }

```

(End of definition for `\hook_show:n` and others. These functions are documented on page 209.)

`\__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 = <label1>|<label2>`, and `#3 = <hook>` (the latter may be the argument `#1` to `\__hook_list_rules:nn`, or `??` if it is a default rule).

```

2074 \cs_new_protected:Npn __hook_list_rules:nn #1 #2
2075 {
2076 \prop_if_exist:cT { g__hook_#1_code_prop }
2077 {
2078 \cs_set_protected:Npn __hook_tmp:w ##1 ##2 ##3 {#2}
2079 \prop_map_inline:cn { g__hook_#1_code_prop }
2080 {
2081 \prop_map_inline:cn { g__hook_#1_code_prop }
2082 {
2083 __hook_if_label_case:nnnn {##1} {####1}
2084 { \prop_map_break: }
2085 { __hook_list_one_rule:nnn {##1} {####1} }
2086 { __hook_list_one_rule:nnn {####1} {##1} }
2087 {#1}
2088 }
2089 }
2090 }
2091 }

```

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>`.

```

2092 \cs_new_protected:Npn __hook_list_one_rule:nnn #1#2#3
2093 {
2094 __hook_list_if_rule_exists:nnnF {#1} {#2} {#3}
2095 { __hook_list_if_rule_exists:nnnF {#1} {#2} { ?? } { } }
2096 }
2097 \cs_new_protected:Npn __hook_list_if_rule_exists:nnnF #1#2#3
2098 {
2099 \if_cs_exist:w g__hook_ #3 _rule_ #1 | #2 _tl \cs_end:
2100 \exp_args:Nv __hook_tmp:w

```

```

2101 { g__hook_ #3 _rule_ #1 | #2 _tl } { #1 | #2 } {#3}
2102 \exp_after:wN \use_none:nn
2103 \fi:
2104 \use:n
2105 }

(End of 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.

```

2106 \cs_new_protected:Npn __hook_debug_print_rules:n #1
2107 {
2108 \iow_term:n { The~hook~#1~contains~the~rules: }
2109 \cs_set_protected:Npn __hook_tmp:w ##1
2110 {
2111 __hook_list_rules:nn {#1}
2112 {
2113 \iow_term:x
2114 {
2115 > ##1 {####2} ##1 => ##1 {####1}
2116 \str_if_eq:nnT {####3} {??} { ~ (default) }
2117 }
2118 }
2119 }
2120 \exp_args:No __hook_tmp:w { \use:nn { ~ } { ~ } }
2121 }
```

(End of definition for \\_\_hook\_debug\_print\_rules:n.)

## 4.8 Specifying code for next invocation

\hook\_gput\_next\_code:nn

```

2122 \langle latexrelease \rangle \IncludeInRelease{2023/06/01}{\hook_gput_next_code:nn}
2123 \langle latexrelease \rangle \cs_new_protected:Npn \hook_gput_next_code:nn #1 #2
2124 {
2125 __hook_replacing_args_false:
2126 __hook_normalize_hook_args:Nn __hook_gput_next_code:nn {#1} {#2}
2127 __hook_replacing_args_reset:
2128 }
2129 \cs_new_protected:Npn \hook_gput_next_code_with_args:nn #1 #2
2130 {
2131 __hook_replacing_args_true:
2132 __hook_normalize_hook_args:Nn __hook_gput_next_code:nn {#1} {#2}
2133 __hook_replacing_args_reset:
2134 }
2135 \langle latexrelease \rangle \EndIncludeInRelease
2136 \langle latexrelease \rangle \IncludeInRelease{2020/10/01}{\hook_gput_next_code:nn}
2137 \langle latexrelease \rangle \cs_gset_protected:Npn \hook_gput_next_code:nn #1
2138 \langle latexrelease \rangle __hook_normalize_hook_args:Nn
2139 \langle latexrelease \rangle __hook_gput_next_code:nn {#1} }
2140 \langle latexrelease \rangle \cs_gset_protected:Npn \hook_gput_next_code_with_args:nn #1 #2 { }
2141 \langle latexrelease \rangle \EndIncludeInRelease
```

(End of definition for `\_hook_gput_next_code:nn`. This function is documented on page 208.)

```
_hook_gput_next_code:nn
2144 \cs_new_protected:Npn _hook_gput_next_code:nn #1 #2
2145 {
2146 _hook_if_disabled:nTF {#1}
2147 { \msg_error:nnn { hooks } { hook-disabled } {#1} }
2148 {
2149 _hook_if_structure_exist:nTF {#1}
2150 { _hook_gput_next_do:nn }
2151 { _hook_try_declar ing_generic_next_hook:nn }
2152 {#1} {#2}
2153 }
2154 }
```

(End of definition for `\_hook_gput_next_code:nn`.)

`\_hook_gput_next_do:nn` Start by sanity-checking with `\_hook_chk_args_allowed:nn`. Then 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.

```
2155 \begin{textrun}\IncludeInRelease{2023/06/01}{_hook_gput_next_do:nn}
2156 \end{textrun} {Hooks~with~args}
2157 \cs_new_protected:Npn _hook_gput_next_do:nn #1
2158 {
2159 _hook_init_structure:n {#1}
2160 _hook_chk_args_allowed:nn {#1} { AddToHookNext }
2161 _hook_cs_if_empty:cT { _hook~#1 }
2162 { _hook_update_hook_code:n {#1} }
2163 _hook_cs_if_empty:cT { _hook_next~#1 }
2164 { _hook_next_gset:nn {#1} { _hook_clear_next:n {#1} } }
2165 _hook_cs_gput_right:nnn { _next } {#1}
2166 }
2167 \begin{textrun}\EndIncludeInRelease
2168 \end{textrun} {Hooks~with~args}
2169 \begin{textrun}\IncludeInRelease{2020/10/01}{_hook_gput_next_do:nn}
2170 \end{textrun} {
2171 \begin{textrun}\cs_gset_protected:Npn _hook_gput_next_do:nn #1
2172 \end{textrun} {
2173 \begin{textrun}\exp_args:Nc _hook_gput_next_do:Nnn
2174 \end{textrun} { _hook_next~#1 } {#1}
2175 \begin{textrun}\cs_gset_protected:Npn _hook_gput_next_do:Nnn #1 #2
2176 \end{textrun} {
2177 \begin{textrun}\tl_if_empty:cT { _hook~#2 }
2178 \end{textrun} { _hook_update_hook_code:n {#2} }
2179 \begin{textrun}\tl_if_empty:NT #1
2180 \end{textrun} { _hook_tl_gset:Nn #1 { _hook_clear_next:n {#2} } }
2181 \begin{textrun}_hook_tl_gput_right:Nn #1
2182 \end{textrun} {
2183 \begin{textrun}\EndIncludeInRelease
```

(End of definition for `\_\_hook_gput_next_do:nn`.)

`\hook_gclear_next_code:n` Discard anything set up for next invocation of the hook.

```
2184 \cs_new_protected:Npn \hook_gclear_next_code:n #1
2185 { __hook_normalize_hook_args:Nn __hook_clear_next:n {#1} }
```

(End of definition for `\hook_gclear_next_code:n`. This function is documented on page 208.)

`\_\_hook_clear_next:n`

```
2186 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{__hook_clear_next:n}
2187 ⟨latexrelease⟩ {Hooks~with~args}
2188 \cs_new_protected:Npn __hook_clear_next:n #1
2189 { __hook_next_gset:nn {#1} { } }
2190 ⟨latexrelease⟩\EndIncludeInRelease
2191 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{__hook_clear_next:n}
2192 ⟨latexrelease⟩ {Hooks~with~args}
2193 ⟨latexrelease⟩\cs_gset_protected:Npn __hook_clear_next:n #1
2194 ⟨latexrelease⟩ { \cs_gset_eq:cn { __hook_next~#1 } \c_empty_tl }
2195 ⟨latexrelease⟩\EndIncludeInRelease
```

(End of definition for `\_\_hook_clear_next:n`.)

## 4.9 Using the hook

`\hook_use: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 thing internally: they check that the hook exists as given, and if so they use it as quickly as possible.

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 `\t1_if_exist:c` is careful to avoid this.

```
2196 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{\hook_use:n}
2197 ⟨latexrelease⟩ {Hooks~with~args}
2198 \cs_new_protected:Npn \hook_use:n #1
2199 {
2200 __hook_preamble_hook:n {#1}
2201 __hook_use_initialized:n {#1}
2202 }
2203 \cs_new:Npn __hook_use_initialized:n #1
2204 {
2205 \if_cs_exist:w __hook~#1 \cs_end:
2206 \cs:w __hook~#1 \use_i:nn
2207 \fi:
2208 \use_none:n
```

```

2209 \cs_end:
2210 }
2211 \cs_new_protected:Npn __hook_preamble_hook:n #1
2212 {
2213 \if_cs_exist:w __hook~#1 \cs_end:
2214 __hook_initialize_hook_code:n {#1}
2215 \fi:
2216 }
2217 \EndIncludeInRelease
2218 \IncludeInRelease{2021/11/15}{\hook_use:n}
2219 \Standardise-generic-hook-names
2220 \cs_new_protected:Npn \hook_use:n #1
2221 \tlt_if_exist:cT { __hook~#1 }
2222 \tlt_if_exist:cT { __hook~#1 }
2223 __hook_preamble_hook:n {#1}
2224 __hook_use:n #1 \cs_end:
2225 __hook_use:n #1 \cs_end:
2226 __hook_use:n #1 \cs_end:
2227 __hook_use:n #1 \cs_end:
2228 __hook_use:n #1 __hook_use_initialized:n #1
2229 \tlt_if_exist:cT { __hook~#1 \cs_end:
2230 __hook_use:n #1 \exp_after:wN \cs_end:
2231 __hook_use:n #1 \exp_after:wN \cs_end:
2232 __hook_use:n #1 \fi:
2233 __hook_use:n #1 \cs_end:
2234 __hook_use:n #1 __hook_preamble_hook:n #1
2235 __hook_use:n #1 __hook_initialize_hook_code:n {#1} \fi:
2236 __hook_use:n #1 __hook_use:nnw #1 \fi:
2237 \EndIncludeInRelease
2238 \IncludeInRelease{2020/10/01}{\hook_use:n}
2239 \Standardise-generic-hook-names
2240 \cs_new_protected:Npn \hook_use:n #1
2241 \tlt_if_exist:cTF { __hook~#1 }
2242 \tlt_if_exist:cTF { __hook~#1 }
2243 __hook_preamble_hook:n {#1}
2244 __hook_use:n #1 \cs_end:
2245 __hook_use:n #1 \cs_end:
2246 __hook_use:n #1 \cs_end:
2247 __hook_use:wn #1 / \s__hook_mark {#1} \fi:
2248 __hook_use:wn #1 / \s__hook_mark {#1} \fi:
2249 __hook_use:n #1 __hook_use_initialized:n #1
2250 \tlt_if_exist:cT { __hook~#1 \cs_end:
2251 __hook_use:n #1 \else:
2252 __hook_use:n #1 __hook_use_undefined:w
2253 __hook_use:n #1 \fi:
2254 __hook_use:n #1 \cs_end:
2255 __hook_use:n #1 __hook_use_end:
2256 __hook_use:n #1 \cs_end:
2257 __hook_use:n #1 __hook_use_undefined:w
2258 __hook_use:n #1 __hook_use_end:
2259 __hook_use:n #1 __hook_use_end:
2260 __hook_use:n #1 % fi
2261 __hook_use:wn #3 / \s__hook_mark {#3}

```

```

2262 〈latexrelease〉 }
2263 〈latexrelease〉\cs_new_protected:Npn __hook_preamble_hook:n #1
2264 〈latexrelease〉 { __hook_initialize_hook_code:n {#1} }
2265 〈latexrelease〉\cs_new_eq:NN __hook_use_end: \cs_end:
2266 〈latexrelease〉\cs_new:Npn \hook_use:nnw #1 { }
2267 〈latexrelease〉\EndIncludeInRelease

(End of definition for \hook_use:n, __hook_use_initialized:n, and __hook_preamble_hook:n. This function is documented on page 207.)

```

### \hook\_use:nnw

```

__hook_use_initialized:nnw
2268 〈latexrelease〉\IncludeInRelease{2023/06/01}{\hook_use:nnw}
2269 〈latexrelease〉
2270 〈latexrelease〉\cs_new_protected:Npn \hook_use:nnw #1
2271 {
2272 __hook_preamble_hook:n {#1}
2273 __hook_use_initialized:nnw {#1}
2274 }
2275 \cs_new:Npn __hook_use_initialized:nnw #1 #2
2276 {
2277 \cs:w
2278 \if_cs_exist:w __hook~#1 \cs_end:
2279 __hook~#1
2280 \else:
2281 use_none: \prg_replicate:nn {#2} { n }
2282 \fi:
2283 \cs_end:
2284 }
2285 〈latexrelease〉\EndIncludeInRelease
2286 〈latexrelease〉\IncludeInRelease{2020/10/01}{\hook_use:nnw}
2287 〈latexrelease〉
2288 〈latexrelease〉\cs_gset:Npn \hook_use:nnw #1 #2
2289 〈latexrelease〉 { \use:c { use_none: \prg_replicate:nn {#2} { n } } }
2290 〈latexrelease〉\EndIncludeInRelease

```

(*End of definition for \hook\_use:nnw and \\_\_hook\_use\_initialized:nnw. This function is documented on page 207.*)

### \\_\_hook\_post\_initialization\_defs:

```

2291 〈latexrelease〉\IncludeInRelease{2023/06/01}{__hook_post_initialization_defs:}
2292 〈latexrelease〉
2293 〈latexrelease〉\cs_new_protected:Npn __hook_post_initialization_defs:
2294 {
2295 \cs_gset_eq:NN \hook_use:n __hook_use_initialized:n
2296 \cs_gset_eq:NN \hook_use:nnw __hook_use_initialized:nnw
2297 \cs_gset_eq:NN __hook_preamble_hook:n \use_none:n
2298 \cs_gset_eq:NN __hook_post_initialization_defs: \prg_do_nothing:
2299 }
2300 〈latexrelease〉\EndIncludeInRelease
2301 〈latexrelease〉\IncludeInRelease{2020/10/01}{__hook_post_initialization_defs:}
2302 〈latexrelease〉
2303 〈latexrelease〉\cs_undefine:N __hook_post_initialization_defs:
2304 〈latexrelease〉\EndIncludeInRelease

```

(*End of definition for \\_\_hook\_post\_initialization\_defs:.*)

`\_\_hook\_use:wn` `\_\_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_usable_use:n` checks if the hook exist, and calls `\_\_hook_preamble_hook:n` if so, then uses the hook.

```

2305 <|latexrelease>\IncludeInRelease{2021/11/15}{__hook_use:wn}
2306 <|latexrelease> {Standardise-generic-hook-names}
2307 <|latexrelease>\EndIncludeInRelease
2308 <|latexrelease>\IncludeInRelease{2020/10/01}{__hook_use:wn}
2309 <|latexrelease> {Standardise-generic-hook-names}
2310 <|latexrelease>\cs_new:Npn __hook_use:wn #1 / #2 \s__hook_mark #3
2311 <|latexrelease> {
2312 <|latexrelease> \str_if_eq:nnTF {#1} { file }
2313 <|latexrelease> { __hook_try_file_hook:n {#3} }
2314 <|latexrelease> { } % Hook doesn't exist
2315 <|latexrelease> }

2316 <|latexrelease>\cs_new_protected:Npn __hook_try_file_hook:n #
2317 <|latexrelease> {
2318 <|latexrelease> __hook_if_file_hook:wTF #1 / / \s__hook_mark
2319 <|latexrelease> {
2320 <|latexrelease> \exp_args:Ne __hook_if_usable_use:n
2321 <|latexrelease> { \exp_args:Ne __hook_file_hook_normalize:n {#1} }
2322 <|latexrelease> }
2323 <|latexrelease> { __hook_if_usable_use:n {#1} }
2324 <|latexrelease> % file/ generic hook (e.g. file/before)
2325 <|latexrelease> }

2326 <|latexrelease>\cs_new_protected:Npn __hook_if_usable_use:n #
2327 <|latexrelease> {
2328 <|latexrelease> \tl_if_exist:cT { __hook~#1 }
2329 <|latexrelease> {
2330 <|latexrelease> __hook_preamble_hook:n {#1}
2331 <|latexrelease> \cs:w __hook~#1 \cs_end:
2332 <|latexrelease> }
2333 <|latexrelease> }
2334 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\_\_hook_use:wn`, `\_\_hook_try_file_hook:n`, and `\_\_hook_if_usable_use:n`.)

`\hook_use_once:n`  
`\hook_use_once:nnw`

For hooks that can and should be used only once we have a special use command that further inhibits the hook from getting more code added to it. 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.

```

2335 <|latexrelease>\IncludeInRelease{2023/06/01}{\hook_use_once:nnw}
2336 <|latexrelease> {Hooks-with-args}
2337 \cs_new_protected:Npn \hook_use_once:n #
2338 {
2339 __hook_if_execute_immediately:nF {#1}

```

```

2340 { __hook_normalize_hook_args:Nn __hook_use_once:nn
2341 { \use:n {#1} } { 0 } }
2342 }
2343 \cs_new_protected:Npn \hook_use_once:nnw #1 #2
2344 {
2345 __hook_if_execute_immediately:nF {#1}
2346 { __hook_normalize_hook_args:Nn __hook_use_once:nn
2347 { \use:n {#1} } {#2} }
2348 }
2349 \end{IncludeInRelease}

(End of definition for \hook_use_once:n and \hook_use_once:nnw. These functions are documented
on page 207.)
```

```

2350 \end{IncludeInRelease}{2020/10/01}{\hook_use_once:nnw}
2351 \end{IncludeInRelease} {Hooks~with~args}
2352 \end{IncludeInRelease}\cs_gset_protected:Npn \hook_use_once:n #1
2353 \end{IncludeInRelease} {
2354 \end{IncludeInRelease} __hook_if_execute_immediately:nF {#1}
2355 \end{IncludeInRelease} { __hook_normalize_hook_args:Nn __hook_use_once:n
2356 \end{IncludeInRelease} { \use:n {#1} } }
2357 \end{IncludeInRelease} }
2358 \end{IncludeInRelease}\cs_gset:Npn \hook_use_once:nnw #1 #2
2359 \end{IncludeInRelease} { \use:c { use_none: \prg_replicate:nn {#2} { n } } }
2360 \end{IncludeInRelease}\end{IncludeInRelease}

__hook_use_once:nn
```

```

2361 \end{IncludeInRelease}{2023/06/01}{__hook_use_once:nn}
2362 \end{IncludeInRelease} {Hooks~with~args}
2363 \end{IncludeInRelease}\cs_new_protected:Npn __hook_use_once:nn #1 #2
2364 {
2365 __hook_preamble_hook:n {#1}
2366 __hook_use_once_set:n {#1}
```

When a hook has arguments, the call to \\_\_hook\_use\_initialized:n, should be the very last thing to happen, otherwise the arguments grabbed will be wrong. So, to clean up after the hook we need to cheat a bit and sneak the cleanup code at the end of the hook, along with the next execution code.

```

2367 __hook_replacing_args_false:
2368 __hook_cs_gput_right:nnn { _next } {#1}
2369 { __hook_use_once_clear:n {#1} }
2370 __hook_replacing_args_reset:
2371 __hook_if_usable:nTF {#1}
2372 { __hook_use_initialized:n {#1} }
2373 {
2374 \int_compare:nNnT {#2} > { 0 }
2375 { \use:c { use_none: \prg_replicate:nn {#2} { n } } }
2376 }
2377 }
2378 \end{IncludeInRelease}
2379 %
2380 \end{IncludeInRelease}{2020/10/01}{__hook_use_once:nn}
2381 \end{IncludeInRelease} {Hooks~with~args}
2382 \end{IncludeInRelease}\cs_gset_protected:Npn __hook_use_once:n #1
2383 \end{IncludeInRelease} {
```

```

2384 <latexrelease> __hook_preamble_hook:n {#1}
2385 <latexrelease> __hook_use_once_set:n {#1}
2386 <latexrelease> __hook_use_initialized:n {#1}
2387 <latexrelease> __hook_use_once_clear:n {#1}
2388 <latexrelease> }
2389 <latexrelease>\cs_undefine:N __hook_use_once:nn
2390 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\__hook_use_once:nn`.)

`\__hook_use_once_set:n` `\__hook_use_once_clear:n` `\__hook_use_once_set:n` is used before the actual hook code is executed so that any usage of `\AddToHook` inside the hook causes the code to execute immediately. Setting `\g__hook_{hook}_reversed_tl` to `I` prevents further code from being added to the hook. `\__hook_use_once_clear:n` then clears the hook so that any further call to `\hook_use:n` or `\hook_use_once:n` will expand to nothing.

```

2391 <latexrelease>\IncludeInRelease{2023/06/01}{__hook_use_once_clear:n}
2392 <latexrelease> {Hooks~with~args}
2393 \cs_new_protected:Npn __hook_use_once_set:n #1
2394 { __hook_tl_gset:cn { g__hook_#1_reversed_tl } { I } }
2395 \cs_new_protected:Npn __hook_use_once_clear:n #1
2396 {
2397 __hook_code_gset:nn {#1} { }
2398 __hook_next_gset:nn {#1} { }
2399 __hook_toplevel_gset:nn {#1} { }
2400 \prop_gclear_new:c { g__hook_#1_code_prop }
2401 }
2402 <latexrelease>\EndIncludeInRelease
2403 <latexrelease>\IncludeInRelease{2020/10/01}{__hook_use_once_clear:n}
2404 <latexrelease> {Hooks~with~args}
2405 \cs_new_protected:Npn __hook_use_once_clear:n #1
2406 <latexrelease> {
2407 __hook_tl_gclear:c { __hook~#1 }
2408 __hook_tl_gclear:c { __hook_next~#1 }
2409 __hook_tl_gclear:c { __hook_toplevel~#1 }
2410 \prop_gclear_new:c { g__hook_#1_code_prop }
2411 <latexrelease> }
2412 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\__hook_use_once_set:n` and `\__hook_use_once_clear:n`.)

`\__hook_if_execute_immediately_p:n` `\__hook_if_execute_immediately:nTF` To check whether the code being added should be executed immediately (that is, if the hook is a one-time hook), we check if `\g__hook_{hook}_reversed_tl` is `I`. The gymnastics around `\if:w` is there to allow the `reversed` token list to be empty.

```

2413 \prg_new_conditional:Npnn __hook_if_execute_immediately:n #1 { T, F, TF }
2414 {
2415 \exp_after:wN __hook_use_none_delimit_by_s_mark:w
2416 \if:w I
2417 \if_cs_exist:w g__hook_#1_reversed_tl \cs_end:
2418 \cs:w g__hook_#1_reversed_tl \exp_after:wN \cs_end:
2419 \fi:
2420 X
2421 \s__hook_mark \prg_return_true:
2422 \else:
2423 \s__hook_mark \prg_return_false:

```

```

2424 \fi:
2425 }

```

(End of definition for `\_hook_if_execute_immediately:nTF`.)

## 4.10 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 several other states as discussed in 4.4.2. 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) or may be disabled.

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).

All other states can only be queried with internal tests as the different states are irrelevant for package code.

```

\hook_if_empty_p:n Test if a hook is empty (that is, no code was added to that hook). A <hook> being empty
\hook_if_empty:nTF means that all three of its \g__hook_<hook>_code_prop, its _hook_toplevel_<hook> and its _hook_next_<hook> are empty.
2426 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{\hook_if_empty:n}
2427 ⟨latexrelease⟩ {Hooks~with-args}
2428 \prg_new_conditional:Npnn \hook_if_empty:n #1 { p , T , F , TF }
2429 {
2430 \if:w
2431 T
2432 \prop_if_exist:cT { g__hook_#1_code_prop }
2433 { \prop_if_empty:cF { g__hook_#1_code_prop } { F } }
2434 _hook_cs_if_empty:cF { __hook_toplevel~#1 } { F }
2435 _hook_cs_if_empty:cF { __hook_next~#1 } { F }
2436 T
2437 \prg_return_true:
2438 \else:
2439 \prg_return_false:
2440 \fi:
2441 }
2442 ⟨latexrelease⟩\EndIncludeInRelease
2443 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\hook_if_empty:n}
2444 ⟨latexrelease⟩ {Hooks~with-args}
2445 ⟨latexrelease⟩\prg_new_conditional:Npnn \hook_if_empty:n #1 { p , T , F , TF }
2446 ⟨latexrelease⟩ {
2447 ⟨latexrelease⟩ _hook_if_structure_exist:nTF {#1}
2448 ⟨latexrelease⟩ {
2449 ⟨latexrelease⟩ \bool_lazy_and:nnTF
2450 ⟨latexrelease⟩ { \prop_if_empty_p:c { g__hook_#1_code_prop } }
2451 ⟨latexrelease⟩ {
2452 ⟨latexrelease⟩ \bool_lazy_and_p:nn
2453 ⟨latexrelease⟩ { \tl_if_empty_p:c { __hook_toplevel~#1 } }

```

```

2454 <|latexrelease> { \tl_if_empty_p:c { __hook_next~#1 } }
2455 <|latexrelease> }
2456 <|latexrelease> { \prg_return_true: }
2457 <|latexrelease> { \prg_return_false: }
2458 <|latexrelease> }
2459 <|latexrelease> { \prg_return_true: }
2460 <|latexrelease> }
2461 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\hook_if_empty:nTF`. This function is documented on page 209.)

`\__hook_if_usable_p:n` A hook is usable 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).

```

2462 \prg_new_conditional:Npnn __hook_if_usable:n #1 { p , T , F , TF }
2463 {
2464 \cs_if_exist:cTF { __hook~#1 }
2465 { \prg_return_true: }
2466 { \prg_return_false: }
2467 }

```

(End of definition for `\__hook_if_usable:nTF`.)

`\__hook_if_structure_exist_p:n` An internal check if the hook has already its basic internal structure set up with `\__hook_init_structure: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`.

```

2468 \prg_new_conditional:Npnn __hook_if_structure_exist:n #1 { p , T , F , TF }
2469 {
2470 \prop_if_exist:cTF { g__hook_{#1}_code_prop }
2471 { \prg_return_true: }
2472 { \prg_return_false: }
2473 }

```

(End of definition for `\__hook_if_structure_exist:nTF`.)

`\__hook_if_declared_p:n` Internal test to check if the hook was officially declared with `\hook_new:n` or a variant.

```

2474 \prg_new_conditional:Npnn __hook_if_declared:n #1 { p , T , F , TF }
2475 {
2476 \tl_if_exist:cTF { g__hook_{#1}_declared_tl }
2477 { \prg_return_true: }
2478 { \prg_return_false: }
2479 }

```

(End of definition for `\__hook_if_declared:nTF`.)

`\__hook_if_reversed_p:n` An internal conditional that checks if a hook is reversed.

```

2480 \prg_new_conditional:Npnn __hook_if_reversed:n #1 { p , T , F , TF }
2481 {
2482 \exp_after:wN __hook_use_none_delimit_by_s_mark:w
2483 \if:w - \cs:w g__hook_{#1}_reversed_tl \cs_end:
2484 \s__hook_mark \prg_return_true:
2485 \else:

```

```

2486 \s__hook_mark \prg_return_false:
2487 \fi:
2488 }
```

(End of definition for `\_hook_if_reversed:nTF`.)

`\_hook_if_generic_p:n` An internal conditional that checks if a name belongs to a generic hook. The deprecated version needs to check if #3 is empty to avoid returning true on `file/before`, for example.

```

2489 \prg_new_conditional:Npnn _hook_if_generic:n #1 { T, TF }
2490 { _hook_if_generic:w #1 / / / \s__hook_mark }
2491 \cs_new:Npn _hook_if_generic:w #1 / #2 / #3 / #4 \s__hook_mark
2492 {
2493 \cs_if_exist:cTF { c__hook_generic:#1/./#3_t1 }
2494 { \prg_return_true: }
2495 { \prg_return_false: }
2496 }
2497 \prg_new_conditional:Npnn _hook_if_deprecated_generic:n #1 { T, TF }
2498 { _hook_if_deprecated_generic:w #1 / / / \s__hook_mark }
2499 \cs_new:Npn _hook_if_deprecated_generic:w #1 / #2 / #3 / #4 \s__hook_mark
2500 {
2501 \cs_if_exist:cTF { c__hook_deprecated:#1/./#2_t1 }
2502 {
2503 \tl_if_empty:nTF {#3}
2504 { \prg_return_false: }
2505 { \prg_return_true: }
2506 }
2507 { \prg_return_false: }
2508 }
```

(End of definition for `\_hook_if_generic:nTF` and `\_hook_if_deprecated_generic:nTF`.)

`\_hook_if_cmd_hook_p:n` An internal conditional that checks if a given hook is a valid generic cmd hook.

```

2509 <|latexrelease>\IncludeInRelease{2023/06/01}{_hook_if_cmd_hook:n}
2510 <|latexrelease> {Hooks~with~args}
2511 \prg_new_conditional:Npnn _hook_if_cmd_hook:n #1 { T }
2512 { _hook_if_cmd_hook:w #1 / / / \s__hook_mark }
2513 \cs_new:Npn _hook_if_cmd_hook:w #1 / #2 / #3 / #4 \s__hook_mark
2514 {
2515 \if:w Y
2516 \str_if_eq:nnF {#1} { cmd } { N }
2517 \tl_if_exist:cF { c__hook_generic:#1/./#3_t1 } { N }
2518 Y
2519 \prg_return_true:
2520 \else:
2521 \prg_return_false:
2522 \fi:
2523 }
2524 <|latexrelease>\EndIncludeInRelease
2525 <|latexrelease>\IncludeInRelease{2020/10/01}{_hook_if_cmd_hook:n}
2526 <|latexrelease> {Hooks~with~args}
2527 <|latexrelease>\cs_undefine:N _hook_if_cmd_hook:nT
2528 <|latexrelease>\EndIncludeInRelease
```

(End of definition for `\_hook_if_cmd_hook:nTF` and `\_hook_if_cmd_hook:wTF`.)

`\_hook_if_generic_reversed_p:n` An internal conditional that checks if a name belongs to a generic reversed hook.

```

2529 \prg_new_conditional:Npnn _hook_if_generic_reversed:n #1 { T }
2530 { _hook_if_generic_reversed:w #1 / / \scan_stop: }
2531 \cs_new:Npn _hook_if_generic_reversed:w #1 / #2 / #3 / #4 \scan_stop:
2532 {
2533 \if_charcode:w - \cs:w c_hook_generic_#1./.#3_tl \cs_end:
2534 \prg_return_true:
2535 \else:
2536 \prg_return_false:
2537 \fi:
2538 }
```

(End of definition for `\_hook_if_generic_reversed:nTF`.)

`\_hook_if_replacing_args:TF` An internal conditional that checks if the code being added to the hook contains arguments.

```

_hook_misused_if_replacing_args:nn
_hook_replacing_args_true:
 _hook_replacing_args_false:
 _hook_replacing_args_reset:
\g_hook_replacing_stack_seq
2539 \seq_new:N \g_hook_replacing_stack_seq
2540 \cs_new:Npn _hook_misused_if_replacing_args:nn #1 #2
2541 {
2542 \msg_expandable_error:nnn { latex2e } { should-not-happen }
2543 { Misused-_hook_if_replacing_args::. }
2544 }
2545 \cs_new:Npn _hook_if_replacing_args:TF
2546 { _hook_misused_if_replacing_args:nn }
2547 \cs_new_protected:Npn _hook_replacing_args_true:
2548 {
2549 \seq_gpush:No \g_hook_replacing_stack_seq
2550 { _hook_if_replacing_args:TF }
2551 \cs_set:Npn _hook_if_replacing_args:TF { \use_i:nn }
2552 }
2553 \cs_new_protected:Npn _hook_replacing_args_false:
2554 {
2555 \seq_gpush:No \g_hook_replacing_stack_seq
2556 { _hook_if_replacing_args:TF }
2557 \cs_set:Npn _hook_if_replacing_args:TF { \use_ii:nn }
2558 }
2559 \cs_new_protected:Npn _hook_replacing_args_reset:
2560 {
2561 \seq_gpop:NN \g_hook_replacing_stack_seq \l_hook_return_tl
2562 \cs_gset_eq:NN _hook_if_replacing_args:TF \l_hook_return_tl
2563 }
```

(End of definition for `\_hook_if_replacing_args:TF` and others.)

## 4.11 Messages

Hook errors are LaTeX kernel errors:

```

2564 \prop_gput:Nnn \g_msg_module_type_prop { hooks } { LaTeX }
And so are kernel errors (this should move elsewhere eventually).
```

```

2565 \prop_gput:Nnn \g_msg_module_type_prop { latex2e } { LaTeX }
2566 \prop_gput:Nnn \g_msg_module_name_prop { latex2e } { kernel }
```

```

2567 \msg_new:nnnn { hooks } { labels-incompatible }
2568 {
2569 Labels-'#1'~and-'#2'~are~incompatible
2570 \str_if_eq:nnF {#3} {??} { ~in~hook-'#3' } .~
2571 \int_compare:nNnTF {#4} = { 1 }
2572 { The~ code~ for~ both~ labels~ will~ be~ dropped. }
2573 { You~ may~ see~ errors~ later. }
2574 }
2575 { LaTeX-found~two~incompatible~labels~in~the~same~hook.~
2576 This~indicates~an~incompatibility~between~packages. }
2577 \msg_new:nnnn { hooks } { exists }
2578 { Hook-'#1'~ has~ already~ been~ declared. }
2579 { There~ already~ exists~ a~ hook~ declaration~ with~ this~
2580 name.\\
2581 Please~ use~ a~ different~ name~ for~ your~ hook.}
2582 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{too-many-args}
2583 ⟨latexrelease⟩ {Hooks-with-args}

2584 \msg_new:nnnn { hooks } { too-many-args }
2585 { Too~many~arguments~for~hook-'#1'. }
2586 {
2587 You~tried~to~declare~a~hook~with~#2~arguments,~but~a~
2588 hook~can~only~have~up~to~nine.~LaTeX~will~define~this~
2589 hook~with~nine~arguments.
2590 }
2591 \msg_new:nnnn { hooks } { without-args }
2592 { Hook-'#1'~has~no~arguments. }
2593 {
2594 You~tried~to~use~\iow_char:N\#2WithArguments~
2595 on~a~hook~that~takes~no~arguments.\\
2596 Check~the~usage~of~the~hook~or~use~\iow_char:N\#2~instead.\\
2597 \\
2598 LaTeX~will~use~\iow_char:N\#2.
2599 }
2600 \msg_new:nnnn { hooks } { one-time-args }
2601 { You~can't~have~arguments~in~used~one-time~hook-'#1'. }
2602 {
2603 You~tried~to~use~\iow_char:N\#2WithArguments~
2604 on~a~one-time~hook~that~has~already~been~used.~
2605 You~have~to~add~the~code~before~the~hook~is~used,~
2606 or~add~the~code~without~arguments~using~\iow_char:N\#2~instead.\\
2607 \\
2608 LaTeX~will~use~\iow_char:N\#2.
2609 }
2610 ⟨latexrelease⟩\EndIncludeInRelease
2611 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{too-many-args}
2612 ⟨latexrelease⟩ {Hooks-with-args}
2613 ⟨latexrelease⟩\EndIncludeInRelease
2614 \msg_new:nnnn { hooks } { hook-disabled }
2615 { Cannot~add~code~to~disabled~hook-'#1'. }
2616 {
2617 The~hook-'#1'~you~tried~to~add~code~to~was~previously~disabled~
```

```

2618 with~\iow_char:N\\hook_disable_generic:n~or~
2619 \iow_char:N\\DisableGenericHook,~so~
2620 it~cannot~have~code~added~to~it.
2621 }
2622 \\msg_new:nnn { hooks } { empty-label }
2623 {
2624 Empty~code~label~\\msg_line_context:.~
2625 Using~'__hook_currname_or_default:'~instead.
2626 }
2627 \\msg_new:nnn { hooks } { empty-hook }
2628 {
2629 Empty~hook~name~\\msg_line_context:.
2630 }
2631 \\msg_new:nnn { hooks } { no-default-label }
2632 {
2633 Missing~(empty)~default~label~\\msg_line_context:. \\
2634 This~command~was~ignored.
2635 }
2636 \\msg_new:nnnn { hooks } { unknown-rule }
2637 {
2638 Unknown~ relationship~ '#3'~
2639 between~ labels~ '#2'~ and~ '#4'~
2640 \\str_if_eq:nnF {#1} {??} { ~in~hook~'#1' }. ~
2641 Perhaps~ a~ misspelling?
2642 }
2643 {
2644 The~ relation~ used~ not~ known~ to~ the~ system.~ Allowed~ values~ are~
2645 'before'~ or~ '<',~
2646 'after'~ or~ '>',~
2647 'incompatible-warning',~
2648 'incompatible-error',~
2649 'voids'~ or~
2650 'unrelated'.
2651 }
2652 \\msg_new:nnnn { hooks } { rule-too-late }
2653 {
2654 Sorting~rule~for~'#1'~hook~applied~too~late.\\\
2655 Try~setting~this~rule~earlier.
2656 }
2657 {
2658 You~tried~to~set~the~ordering~of~hook~'#1'~using\\\
2659 \\iow_char:N\\DeclareHookRule{#1}{#2}{#3}{#4}\\\
2660 but~hook~'#1'~was~already~used~as~a~one-time~hook,~
2661 thus~sorting~is\\\
2662 no~longer~possible.~Declare~the~rule~
2663 before~the~hook~is~used.
2664 }
2665 \\msg_new:nnnn { hooks } { misused-top-level }
2666 {
2667 Illegal~use~of~\\iow_char:N \\AddToHook{#1}[top-level]{...}.\\\
2668 'top-level'~is~reserved~for~the~user's~document.

```

```

2669 }
2670 {
2671 The~'top-level'~label~is~meant~for~user~code~only,~and~should~only~
2672 be~used~(sparingly)~in~the~main~document.~Use~the~default~label~
2673 '__hook_currname_or_default:'~for~this~@\cls@pkg,~or~another~
2674 suitable~label.
2675 }
2676 \msg_new:nnn { hooks } { set-top-level }
2677 {
2678 You~cannot~change~the~default~label~#1~'top-level'.~Illegal \\
2679 \use:nn { ~ } { ~ } \iow_char:N \\#2{#3} \\
2680 \msg_line_context:.
2681 }
2682 \msg_new:nnn { hooks } { extra-pop-label }
2683 {
2684 Extra~\iow_char:N \\PopDefaultHookLabel. \\
2685 This~command~will~be~ignored.
2686 }
2687 \msg_new:nnn { hooks } { missing-pop-label }
2688 {
2689 Missing~\iow_char:N \\PopDefaultHookLabel. \\
2690 The~label~#1~was~pushed~but~never~popped.~Something~is~wrong.
2691 }
2692 \msg_new:nnn { latex2e } { should-not-happen }
2693 {
2694 This~should~not~happen.~#1 \\
2695 Please~report~at~https://github.com/latex3/latex2e.
2696 }
2697 \msg_new:nnn { hooks } { activate-disabled }
2698 {
2699 Cannot~ activate~ hook~ '#1'~ because~ it~ is~ disabled!
2700 }
2701 \msg_new:nnn { hooks } { cannot-remove }
2702 {
2703 Cannot~remove~chunk~#2~from~hook~#1~because~
2704 __hook_if_structure_exist:nTF {#1}
2705 { it~does~not~exist~in~that~hook. }
2706 { the~hook~does~not~exist. }
2707 }
2708 \msg_new:nnn { hooks } { generic-deprecated }
2709 {
2710 Generic~hook~#1/#2/#3~is~deprecated. \\
2711 Use~hook~#1/#3/#2~instead.
2712 }

```

## 4.12 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\epsilon$</sub> package interface commands

**\NewHook** Declaring new hooks ...
  
**\NewReversedHook** 2713 \NewDocumentCommand \NewHook { m }
  
**\NewMirroredHookPair** 2714 { \hook\_new:n {#1} }
  
2715 \NewDocumentCommand \NewReversedHook { m }

```

2716 { \hook_new_reversed:n {#1} }
2717 \NewDocumentCommand \NewMirroredHookPair { mm }
2718 { \hook_new_pair:nn {#1}{#2} }

```

(End of definition for `\NewHook`, `\NewReversedHook`, and `\NewMirroredHookPair`. These functions are documented on page 194.)

### `\NewHookWithArguments`

```

\NewReversedHookWithArguments
\NewMirroredHookPairWithArguments

```

Declaring new hooks with arguments...

```

2719 <latexrelease>\IncludeInRelease{2023/06/01}{\NewHookWithArguments}
2720 <latexrelease> {Hooks~with~args}
2721 \NewDocumentCommand \NewHookWithArguments { mm }
2722 { \hook_new_with_args:nn {#1} {#2} }
2723 \NewDocumentCommand \NewReversedHookWithArguments { mm }
2724 { \hook_new_reversed_with_args:nn {#1} {#2} }
2725 \NewDocumentCommand \NewMirroredHookPairWithArguments { mmm }
2726 { \hook_new_pair_with_args:nnn {#1} {#2} {#3} }
2727 <latexrelease>\EndIncludeInRelease
2728 <latexrelease>\IncludeInRelease{2020/10/01}{\NewHookWithArguments}
2729 <latexrelease> {Hooks~with~args}
2730 <latexrelease>\cs_new_protected:Npn \NewHookWithArguments #1 #2 { }
2731 <latexrelease>\cs_new_protected:Npn \NewReversedHookWithArguments #1 #2 { }
2732 <latexrelease>\cs_new_protected:Npn \NewMirroredHookPairWithArguments #1 #2 #3{ }
2733 <latexrelease>\EndIncludeInRelease

```

(End of definition for `\NewHookWithArguments`, `\NewReversedHookWithArguments`, and `\NewMirroredHookPairWithArguments`. These functions are documented on page 195.)

```

2734 <latexrelease>\IncludeInRelease{2021/06/01}{\ActivateGenericHook}
2735 <latexrelease> {Providing~hooks}

```

### `\ActivateGenericHook`

Providing new hooks ...

```

2736 \NewDocumentCommand \ActivateGenericHook { m }
2737 { \hook_activate_generic:n {#1} }

```

(End of definition for `\ActivateGenericHook`. This function is documented on page 196.)

### `\DisableGenericHook`

Disabling a generic hook.

```

2738 \NewDocumentCommand \DisableGenericHook { m }
2739 { \hook_disable_generic:n {#1} }

```

(End of definition for `\DisableGenericHook`. This function is documented on page 195.)

```

2740 <latexrelease>\EndIncludeInRelease
2741 <latexrelease>\IncludeInRelease{2020/10/01}{\ActivateGenericHook}
2742 <latexrelease> {Providing~hooks}
2743 <latexrelease>\def \ActivateGenericHook #1 { }
2744 <latexrelease>\def \DisableGenericHook #1 { }
2745 <latexrelease>\EndIncludeInRelease

```

### `\AddToHook`

### `\AddToHookWithArguments`

```

2746 <latexrelease>\IncludeInRelease{2023/06/01}{\AddToHookWithArguments}
2747 <latexrelease> {Hooks~with~args}
2748 \NewDocumentCommand \AddToHook { mo +m }
2749 { \hook_gput_code:nnn {#1} {#2} {#3} }
2750 \NewDocumentCommand \AddToHookWithArguments { mo +m }
2751 { \hook_gput_code_with_args:nnn {#1} {#2} {#3} }

```

```

2752 ⟨latexrelease⟩\EndIncludeInRelease
2753 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\AddToHookWithArguments}
2754 ⟨latexrelease⟩ {Hooks~with~args}
2755 ⟨latexrelease⟩\cs_new_protected:Npn \AddToHookWithArguments #1 #2 #3 { }
2756 ⟨latexrelease⟩\EndIncludeInRelease

(End of definition for \AddToHook and \AddToHookWithArguments. These functions are documented on
page 197.)

```

#### \AddToHookNext

\AddToHookNextWithArguments

```

2757 ⟨latexrelease⟩\IncludeInRelease{2023/06/01}{\AddToHookNextWithArguments}
2758 ⟨latexrelease⟩ {Hooks~with~args}
2759 \NewDocumentCommand \AddToHookNext { m +m }
2760 { \hook_gput_next_code:nn {#1} {#2} }
2761 \NewDocumentCommand \AddToHookNextWithArguments { m +m }
2762 { \hook_gput_next_code_with_args:nn {#1} {#2} }
2763 ⟨latexrelease⟩\EndIncludeInRelease
2764 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\AddToHookNextWithArguments}
2765 ⟨latexrelease⟩ {Hooks~with~args}
2766 ⟨latexrelease⟩\cs_new_protected:Npn \AddToHookNextWithArguments #1 #2 { }
2767 ⟨latexrelease⟩\EndIncludeInRelease

(End of definition for \AddToHookNext and \AddToHookNextWithArguments. These functions are
documented on page 199.)

```

#### \ClearHookNext

```

2768 \NewDocumentCommand \ClearHookNext { m }
2769 { \hook_gclear_next_code:n {#1} }

```

(End of definition for \ClearHookNext. This function is documented on page 199.)

#### \RemoveFromHook

```

2770 \NewDocumentCommand \RemoveFromHook { m o }
2771 { \hook_gremove_code:nn {#1} {#2} }

```

(End of definition for \RemoveFromHook. This function is documented on page 198.)

\SetDefaultHookLabel Now define a wrapper that replaces the top of the stack with the argument, and updates \g\_\_hook\_hook\_curr\_name\_tl accordingly.

\PushDefaultHookLabel

```

2772 \NewDocumentCommand \SetDefaultHookLabel { m }
2773 { __hook_set_default_hook_label:n {#1} }

```

The label is only automatically updated with \onefilewithoptions (\usepackage and \documentclass), but some packages, like TikZ, define package-like interfaces, like \usetikzlibrary that are wrappers around \input, so they inherit the default label currently in force (usually `top-level`, but it may change if loaded in another package). To provide a package-like behavior also for hooks in these files, we provide high-level access to the default label stack.

```

2774 \NewDocumentCommand \PushDefaultHookLabel { m }
2775 { __hook_curr_name_push:n {#1} }
2776 \NewDocumentCommand \PopDefaultHookLabel { }
2777 { __hook_curr_name_pop: }

```

The current label stack holds the labels for all files but the current one (more or less like `\@currnamestack`), and the current label token list, `\g__hook_hook_curr_name_tl`, holds the label for the current file. However `\@pushfilename` happens before `\@currname` is set, so we need to look ahead to get the `\@currname` for the label. `expl3` also requires the current file in `\@pushfilename`, so here we abuse `\@expl@push@filename@aux@@` to do `\_hook_curr_name_push:n`.

```
2778 \cs_gset_protected:Npn \@expl@push@filename@aux@@ #1#2#3
2779 {
2780 _hook_curr_name_push:n {#3}
2781 \str_gset:Nx \g_file_curr_name_str {#3}
2782 #1 #2 {#3}
2783 }
```

(End of definition for `\SetDefaultHookLabel`, `\PushDefaultHookLabel`, and `\PopDefaultHookLabel`. These functions are documented on page 202.)

**\UseHook** Avoid the overhead of `xparse` and its protection that we don't want here (since the hook should vanish without trace if empty)!

```
\UseOneTimeHook
\UseHookWithArguments
\UseOneTimeHookWithArguments
2784 <latexrelease>\IncludeInRelease{2023/06/01}{\UseHookWithArguments}
2785 <latexrelease> {Hooks-with-args}
2786 \cs_new:Npn \UseHook { \hook_use:n }
2787 \cs_new:Npn \UseOneTimeHook { \hook_use_once:n }
2788 \cs_new:Npn \UseHookWithArguments { \hook_use:nnw }
2789 \cs_new:Npn \UseOneTimeHookWithArguments { \hook_use_once:nnw }
2790 <latexrelease>\EndIncludeInRelease
2791 <latexrelease>\IncludeInRelease{2020/10/01}{\UseHookWithArguments}
2792 <latexrelease> {Hooks-with-args}
2793 <latexrelease>\cs_new:Npn \UseHookWithArguments #1 #2 { }
2794 <latexrelease>\cs_new:Npn \UseOneTimeHookWithArguments #1 #2 { }
2795 <latexrelease>\EndIncludeInRelease
```

(End of definition for `\UseHook` and others. These functions are documented on page 196.)

```
\ShowHook
\LogHook
2796 \cs_new_protected:Npn \ShowHook { \hook_show:n }
2797 \cs_new_protected:Npn \LogHook { \hook_log:n }
```

(End of definition for `\ShowHook` and `\LogHook`. These functions are documented on page 205.)

```
\DebugHooksOn
\DebugHooksOff
2798 \cs_new_protected:Npn \DebugHooksOn { \hook_debug_on: }
2799 \cs_new_protected:Npn \DebugHooksOff { \hook_debug_off: }
```

(End of definition for `\DebugHooksOn` and `\DebugHooksOff`. These functions are documented on page 206.)

**\DeclareHookRule**

```
2800 \NewDocumentCommand \DeclareHookRule { m m m m }
2801 { \hook_gset_rule:nnnn {#1}{#2}{#3}{#4} }
```

(End of definition for `\DeclareHookRule`. This function is documented on page 203.)

**\DeclareDefaultHookRule** This declaration is only supported before `\begin{document}`.

```
2802 \NewDocumentCommand \DeclareDefaultHookRule { m m m }
2803 { \hook_gset_rule:nnnn {??}{#1}{#2}{#3} }
2804 \@onlypreamble\DeclareDefaultHookRule
```

(End of definition for \DeclareDefaultHookRule. This function is documented on page 204.)

- \ClearHookRule A special setup rule that removes an existing relation. Basically @@\_rule\_gclear:nnn plus fixing the property list for debugging.

*FMi: Needs perhaps an L3 interface, or maybe it should get dropped?*

```
2805 \NewDocumentCommand \ClearHookRule { m m m }
2806 { \hook_gset_rule:nnnn {#1}{#2}{unrelated}{#3} }
```

(End of definition for \ClearHookRule. This function is documented on page 203.)

- \IfHookEmptyTF Here we avoid the overhead of `xparse`, since \IfHookEmptyTF is used in \end (that is, every L<sup>A</sup>T<sub>E</sub>X environment). As a further optimization, use \let rather than \def to avoid one expansion step.

```
2807 \cs_new_eq:NN \IfHookEmptyTF \hook_if_empty:nTF
2808 \cs_new_eq:NN \IfHookEmptyT \hook_if_empty:nT
2809 \cs_new_eq:NN \IfHookEmptyF \hook_if_empty:nF
```

(End of definition for \IfHookEmptyTF, \IfHookEmptyT, and \IfHookEmptyF. These functions are documented on page 205.)

- \IfHookExistsTF Marked for removal and no longer documented in the doc section!

*PhO: \IfHookExistsTF is used in jlreq.cls, pxatbegshi.sty, pxeverysel.sty, pxeveryshi.sty, so the public name may be an alias of the internal conditional for a while. Regardless, those packages' use for \IfHookExistsTF is not really correct and can be changed.*

```
2810 \cs_new_eq:NN \IfHookExistsTF __hook_if_usable:nTF
```

(End of definition for \IfHookExistsTF.)

## 4.13 Deprecated that needs cleanup at some point

\hook\_disable:n Deprecated.

```
2811 \cs_new_protected:Npn \hook_disable:n
2812 {
2813 __hook_deprecated_warn:nn
2814 { hook_disable:n }
2815 { hook_disable_generic:n }
2816 \hook_disable_generic:n
2817 }
2818 \cs_new_protected:Npn \hook_provide:n
2819 {
2820 __hook_deprecated_warn:nn
2821 { hook_provide:n }
2822 { hook_activate_generic:n }
2823 \hook_activate_generic:n
2824 }
2825 \cs_new_protected:Npn \hook_provide_reversed:n
2826 {
2827 __hook_deprecated_warn:nn
2828 { hook_provide_reversed:n }
2829 { hook_activate_generic:n }
2830 __hook_activate_generic_reversed:n
2831 }
```

```

2832 \cs_new_protected:Npn \hook_provide_pair:nn
2833 {
2834 __hook_DEPRECATED_WARN:nn
2835 { hook_provide_pair:nn }
2836 { hook_activate_generic:n }
2837 __hook_activate_generic_pair:nn
2838 }
2839 \cs_new_protected:Npn __hook_activate_generic_reversed:n #1
2840 { __hook_normalize_hook_args:Nn __hook_activate_generic:nn {#1} { - } }
2841 \cs_new_protected:Npn __hook_activate_generic_pair:nn #1#2
2842 { \hook_activate_generic:n {#1} __hook_activate_generic_reversed:n {#2} }

(End of definition for \hook_disable:n and others.)

```

\DisableHook Deprecated.

```

\ProvideHook
\ProvideReversedHook
\ProvideMirroredHookPair
2843 \cs_new_protected:Npn \DisableHook
2844 {
2845 __hook_DEPRECATED_WARN:nn
2846 { DisableHook }
2847 { DisableGenericHook }
2848 \hook_disable_generic:n
2849 }
2850 \cs_new_protected:Npn \ProvideHook
2851 {
2852 __hook_DEPRECATED_WARN:nn
2853 { ProvideHook }
2854 { ActivateGenericHook }
2855 \hook_activate_generic:n
2856 }
2857 \cs_new_protected:Npn \ProvideReversedHook
2858 {
2859 __hook_DEPRECATED_WARN:nn
2860 { ProvideReversedHook }
2861 { ActivateGenericHook }
2862 __hook_activate_generic_reversed:n
2863 }
2864 \cs_new_protected:Npn \ProvideMirroredHookPair
2865 {
2866 __hook_DEPRECATED_WARN:nn
2867 { ProvideMirroredHookPair }
2868 { ActivateGenericHook }
2869 __hook_activate_generic_pair:nn
2870 }

(End of definition for \DisableHook and others.)

```

\\_\_hook\_DEPRECATED\_WARN:nn Warns about a deprecation, telling what should be used instead.

```

2871 \cs_new_protected:Npn __hook_DEPRECATED_WARN:nn #1 #2
2872 { \msg_warning:nnnn { hooks } { deprecated } {#1} {#2} }
2873 \msg_new:nnn { hooks } { deprecated }
2874 {
2875 Command-\iow_char:N\\#1-is-deprecated-and-will-be-removed-in-a-
2876 future-release. \\ \\
2877 Use-\iow_char:N\\#2-instead.
2878 }

```

(End of definition for `\_\_hook\_deprecated\_warn:nn`.)

## 4.14 Internal commands needed elsewhere

Here we set up a few horrible (but consistent) L<sup>A</sup>T<sub>E</sub>X 2<sub><</sub> 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.

2879 `\_\_hook\_\_=`

`\@expl@@@initialize@all@@`

```
2880 \cs_new_eq:NN \@expl@@@initialize@all@@
2881 __hook_initialize_all:
2882 \cs_new_eq:NN \@expl@@@hook@curr@name@pop@@
2883 __hook_curr_name_pop:
```

(End of definition for `\@expl@@@initialize@all@@` and `\@expl@@@hook@curr@name@pop@@`.)

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.

```
2884 %
2885 \textrun{IncludeInRelease{0000/00/00}{lthooks}}
2886 \textrun{} {The~hook~management}%
2887 \textrun{%
2888 \textrun{\def \NewHook#1{}}
2889 \textrun{\def \NewReversedHook#1{}}
2890 \textrun{\def \NewMirroredHookPair#1#2{}}
2891 \textrun{%
2892 \textrun{\def \DisableGenericHook #1{}}
2893 \textrun{%
2894 \textrun{\long\def \AddToHookNext#1#2{}}
2895 \textrun{%
2896 \textrun{\def \AddToHook#1{\gobble@AddToHook@args}}
2897 \textrun{\providecommand{\gobble@AddToHook@args}[2]{}}
2898 \textrun{%
2899 \textrun{\def \RemoveFromHook#1{\gobble@RemoveFromHook@arg}}
2900 \textrun{\providecommand{\gobble@RemoveFromHook@arg}[1]{}}
2901 \textrun{%
2902 \textrun{\def \UseHook #1{}}
2903 \textrun{\def \UseOneTimeHook #1{}}
2904 \textrun{\def \ShowHook #1{}}
2905 \textrun{\let \DebugHooksOn \empty}
2906 \textrun{\let \DebugHooksOff \empty}
2907 \textrun{%
2908 \textrun{\def \DeclareHookRule #1#2#3#4{}}
2909 \textrun{\def \DeclareDefaultHookRule #1#2#3{}}
2910 \textrun{\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.

```
2911 \textrun{\long\def \IfHookExistsTF #1#2#3{#3}}
2912 \textrun{\long\def \IfHookEmptyTF #1#2#3{#2}}
2913 \textrun{%
2914 \textrun{\EndModuleRelease}
```

```

2915 <@@=hook>
2916 <|latexrelease>\cs:w __hook_rollback_tidyng: \cs_end:
2917 <|latexrelease>\bool_lazy_and:nnt
2918 <|latexrelease> { \int_compare_p:nNn { \sourceLaTeXdate } > { 20230600 } }
2919 <|latexrelease> { \int_compare_p:nNn { \requestedLaTeXdate } < { 20230601 } }
2920 <|latexrelease> {
2921 <|latexrelease> \cs_gset_protected:Npn __hook_rollback_tidyng:
2922 <|latexrelease> {
2923 <|latexrelease> \@latex@error { Rollback~code~executed~twice }
2924 <|latexrelease> {
2925 <|latexrelease> Something~went~wrong~(unless~this~was~
2926 <|latexrelease> done~on~purpose~in~a~testing~environment).
2927 <|latexrelease> }
2928 <|latexrelease> \use_none:nnnn
2929 <|latexrelease> }
2930 <|latexrelease> \cs_set:Npn __hook_tmp:w #1 #2
2931 <|latexrelease> {
2932 <|latexrelease> __hook_tl_gset:cx { __hook#1~#2 }
2933 <|latexrelease> {
2934 <|latexrelease> \exp_args:No \exp_not:o
2935 <|latexrelease> {
2936 <|latexrelease> \cs:w __hook#1~#2 \exp_last_unbraced:N e \cs_end:
2937 <|latexrelease> { __hook_braced_cs_parameter:n
2938 <|latexrelease> { __hook#1~#2 } }
2939 <|latexrelease> }
2940 <|latexrelease> }
2941 <|latexrelease> }
2942 <|latexrelease> \seq_map_inline:Nn \g__hook_all_seq
2943 <|latexrelease> {
2944 <|latexrelease> \exp_after:wN \cs_gset_nopar:Npn
2945 <|latexrelease> \cs:w g__hook_#1_code_prop \exp_args:NNo \exp_args:No
2946 <|latexrelease> \cs_end: { \cs:w g__hook_#1_code_prop \cs_end: }
2947 <|latexrelease> __hook_tmp:w { _toplevel } {#1}
2948 <|latexrelease> __hook_tmp:w { _next } {#1}
2949 <|latexrelease> }
2950 <|latexrelease> }
2951 \ExplSyntaxOff
2952 </2ekernel | latexrelease>
2953 <@@=>

```

# File 09

## ltcmdhooks.dtx

### 1 Introduction

This file implements generic hooks for (arbitrary) commands. In theory every command `\<name>` offers now two associated hooks to which code can be added using `\AddToHook`,<sup>13</sup> `\AddToHookNext`, `\AddToHookWithArguments`, and `\AddToHookNextWithArguments`.<sup>14</sup>

However, this is only true “in theory”. In practice there are a number of restrictions that makes it impossible to use such generic command hooks in a number of cases, so please read all of section 2 to understand what may prevent you from using them successfully.

The generic command hooks are:

`cmd/\<name>/before` This hook is executed at the very start of the command, right after its arguments (if any) are parsed. The hook `<code>` runs in the command inside a call to `\UseHookWithArguments`. Any code added to this hook using `\AddToHookWithArguments` or `\AddToHookNextWithArguments` can access the command’s arguments using #1, #2, etc., up to the number of arguments of the command. If `\AddToHook` or `\AddToHookNext` are used, the arguments cannot be accessed (see the `lthooks` documentation<sup>15</sup> on hooks with arguments).

`cmd/\<name>/after` This hook is similar to `cmd/\<name>/before`, but it is executed at the very end of the command body. This hook is implemented as a reversed hook.

The hooks are not physically present before `\begin{document}`<sup>16</sup> (i.e., using a command in the preamble will never execute the hook) and if nobody has declared any code for them, then they are not added to the command code ever. For example, if we have the following definition

```
\newcommand\foo[2]{Code #1 for #2!}
```

then executing `\foo{A}{B}` will simply run `Code_A_for_B!` as it was always the case. However, if somebody, somewhere (e.g., in a package) adds

```
\AddToHook{cmd/foo/before}{<before code>}
```

then, after `\begin{document}` the definition of `\foo` will be:

```
\renewcommand\foo[2]{%
 \UseHookWithArguments{cmd/foo/before}{2}{#1}{#2}%
 Code #1 for #2!}
```

and similarly `\AddToHook{cmd/foo/after}{<after code>}` alters the definition to

---

<sup>13</sup>In this documentation, when something is being said about `\AddToHook`, the same will be valid for `\AddToHookWithArguments`, unless that particular paragraph is highlighting the differences between both. The same is true for the other hook-related functions and their ...`WithArguments` counterparts.

<sup>14</sup>In practice this is not supported for all types of commands, see section 2.2 for the restrictions that apply and what happens if one tries to use this with commands for which this is not supported.

<sup>15</sup>`texdoc lthooks-doc`

<sup>16</sup>More specifically, they are inserted in the commands after the `\begin{document}` hook, so they are also not present while L<sup>A</sup>T<sub>E</sub>X is reading the `.aux` file.

```
\renewcommand\foo[2]{%
 Code #1 for #2!%
 \UseHookWithArguments{cmd/foo/after}{2}{#1}{#2}}
```

In other words, the mechanism is similar to what `etoolbox` offers with `\preto{cmd}` and `\appto{cmd}` with the important differences

- that code can be prepended or appended (i.e., added to the hooks) even if the command itself is not (yet) defined, because the defining package has not been loaded at this point;
- and that by using the hook management interface it is now possible to define how the code chunks added in these places are ordered, if different packages want to add code at these points.

## 2 Restrictions and Operational details

Adding arbitrary material to commands is tricky because most of the time we do not know what the macro expects as arguments when expanding and `TeX` doesn't have a reliable way to see that, so some guesswork has to be employed.

We can do this in most cases when commands are defined using `\NewDocumentCommand` or `\newcommand` (with a few exceptions). For commands defined with `\def` the situation is less good. Common cases where the command hooks will not work are:

- Commands that use special catcode settings within their definition. In that case it is usually not possible to augment the definition (see [2.1](#)).
- If a command is defined while `\ExplSyntaxOn` is in force **and** the command contains `\~` characters to represent spaces, then it can't be patched to include the command hooks. In fact in some very special circumstances you might even get a low-level error rather than the information that the command can't be patched (see, for example, <https://github.com/latex3/latex2e/issues/1430>).
- Commands that have arguments as far as the user is concerned (e.g., `\section` or `\caption`), but are defined in a way that these arguments are not read by the user level command but only later during the processing. In that case the `after` hook doesn't work at all. The before hook only works with `\AddToHook` but not with `\AddToHookWithArguments` because the arguments haven't been read at that point where the hook is patched in. See section [2.2](#).
- Adding a specific generic command hook is only attempted once per command, thus after redefining a command such hooks will no longer be there and will also not be re-added, see section [2.1.1](#).

All this means that you have to have a good understanding of how commands are defined when you attempt to make use of such hooks and something goes wrong. What can help in that case is to turn on `\DebugHooksOn` in which case you get much more (low-level) details on why something fails and what was tried to enable the hooks.

## 2.1 Patching

The code here tries to find out if a command was defined with `\newcommand` or `\DeclareRobustCommand` or `\NewDocumentCommand`, and if so it *assumes* that the argument specification of the command is as expected (which is not fail-proof, if someone redefines the internals of these commands in devious ways, but is a reasonable assumption).

If the command is one of the defined types, the code here does a sandboxed expansion of the command such that it can be redefined again exactly as before, but with the hook code added.

If however the command is not a known type (it was defined with `\def`, for example), then the code uses an approach similar to `etoolbox`'s `\patchcmd` to retokenize the command with the hook code in place. This procedure, however, is more likely to fail if the catcode settings are not the same as the ones at the time of command's definition, so not always adding a hook to a command will work.

### 2.1.1 Timing

When `\AddToHook` (or its `expl3` equivalent) is called with a generic `cmd` hook, say, `cmd/foo/before`, for the first time (that is, no code was added to that same hook before), in the preamble of a document, it will store a patch instruction for that command until `\begin{document}`, and only then all the commands which had hooks added will be patched in one go. That means that no command in the preamble will have hooks patched into them.

At `\begin{document}` all the delayed patches will be executed, and if the command doesn't exist the code is still added to the hook, but it will not be executed. After `\begin{document}`, when `\AddToHook` is called with a generic `cmd` hook the first time, the command will be immediately patched to include the hook, and if it doesn't exist or if it can't be patched for any reason, an error is thrown; if `\AddToHook` was already used in the preamble no new patching is attempted.

This has the consequence that a command defined or redefined after `\begin{document}` only uses generic `cmd` hook code if `\AddToHook` is called for the first time after the definition is made, or if the command explicitly uses the generic hook in its definition by declaring it with `\NewHookPair` adding `\UseHook` as part of the code.<sup>17</sup>

## 2.2 Commands that look ahead

Some commands are defined in different “steps” and they look ahead in the input stream to find more arguments. If you try to add some code to the `cmd/(name)/after` hook of such command, it will not work, and it is not possible to detect that programmatically, so the user has to know (or find out) which commands can or cannot have hooks attached to them.

One good example is the `\section` command. You can add something to the `cmd/section/before` hook (but only with `\AddToHook` not `\AddToHookWithArguments`), but if you try to add anything to the `cmd/section/after` hook, `\section` will no longer work at all. That happens because the `\section` macro takes no argument, but instead calls a few internal L<sup>A</sup>T<sub>E</sub>X macros to look for the optional and mandatory arguments. By adding code to the `cmd/section/after` hook, you get in the way of that scanning.

---

<sup>17</sup>We might change this behavior in the main document slightly after gaining some usage experience.

In such a case, where it is known that a specific generic command hook does not work if code is added to it, the package author can add a `\DisableGenericHook`<sup>18</sup> declaration to prevent this from happening in user documents and thereby avoiding obscure errors.

### 3 Package Author Interface

The `cmd` hooks are, by default, available for all commands that can be patched to add the hooks. For some commands, however, the very beginning or the very end of the code is not the best place to put the hooks, for example, if the command looks ahead for arguments (see section 2.2).

If you are a package author and you want to add the hooks to your own commands in the proper position you can define the command and manually add the `\UseHookWithArguments` calls inside the command in the proper positions, and manually define the hooks with `\NewHookWithArguments` or `\NewReversedHookWithArguments`. When the hooks are explicitly defined, patching is not attempted so you can make sure your command works properly. For example, an (admittedly not really useful) command that typesets its contents in a framed box with width optionally given in parentheses:

```
\newcommand\fancybox{\@ifnextchar({\@fancybox}{\@fancybox(5cm)}}
\def\@fancybox(#1)#2{\fbox{\parbox{#1}{#2}}}
```

If you try that definition, then add some code after it with

```
\AddToHook{cmd/fancybox/after}{<code>}
```

and then use the `\fancybox` command you will see that it will be completely broken, because the hook will get executed in the middle of parsing for optional (...) argument.

If, on the other hand, you want to add hooks to your command you can do something like:

```
\newcommand\fancybox{\@ifnextchar({\@fancybox}{\@fancybox(5cm)}}
\def\@fancybox(#1)#2{\fbox{%
 \UseHookWithArguments{cmd/fancybox/before}{2}{#1}{#2}%
 \parbox{#1}{#2}%
 \UseHookWithArguments{cmd/fancybox/after}{2}{#1}{#2}}}
\NewHookWithArguments{cmd/fancybox/before}{2}
\NewReversedHookWithArguments{cmd/fancybox/after}{2}
```

then the hooks will be executed where they should and no patching will be attempted. It is important that the hooks are declared with `\NewHookWithArguments` or `\NewReversedHookWithArguments`, otherwise the command hook code will try to patch the command. Note also that the call to `\UseHookWithArguments{cmd/fancybox/before}` does not need to be in the definition of `\fancybox`, but anywhere it makes sense to insert it (in this case in the internal `\@fancybox`).

Alternatively, if for whatever reason your command does not support the generic hooks provided here, you can disable a hook with `\DisableGenericHook`<sup>19</sup>, so that when someone tries to add code to it they will get an error. Or if you don't want the error, you can simply declare the hook with `\NewHook` and never use it.

---

<sup>18</sup>Please use `\DisableGenericHook` if at all, only on hooks that you “own”, i.e., for commands your package or class defines and not second guess whether or not hooks of other packages should get disabled!

<sup>19</sup>Please use `\DisableGenericHook` if at all, only on hooks that you “own”, i.e., for commands your package or class defines and not second guess whether or not hooks of other packages should get disabled!

The above approach is useful for really complex commands where for one or the other reason the hooks can't be placed at the very beginning and end of the command body and some hand-crafting is needed. However, in the example above the real (and in fact only) issue is the cascading argument parsing in the style developed long ago in L<sup>A</sup>T<sub>E</sub>X 2.09. Thus, a much simpler solution for this case is to replace it with the modern `\NewDocumentCommand` syntax and define the command as follows:

```
\DeclareDocumentCommand\fancybox{D(){5cm}m}{\fbox{\parbox{#1}{#2}}}
```

If you do that then both hooks automatically work and are patched into the right places.

### 3.1 Arguments and redefining commands

The code in `ltcmdhooks` does its best to find out how many arguments a given command has, and to insert the appropriate call to `\UseHookWithArguments`, so that the arguments seen by the hook are exactly those grabbed by the command (the hook, after all, is a macro call, so the arguments have to be placed in the right order, or they won't match).

When using the package writer interface, as discussed in section 3, to change the position of the hooks in your commands, you are also free to change how the hook code in your command sees its arguments. When a `cmd` hook is declared with `\NewHook` (or `\NewHookWithArguments` or other variations of that), it loses its "generic" nature and works as a regular hook. This means that you may choose to declare it without arguments regardless if the command takes arguments or not, or declare it with arguments, even if the command takes none.

However, this flexibility should not be abused. When using a nonstandard configuration for the hook arguments, think reasonably: a user will expect that the argument `#1` in the hook corresponds to the argument's first argument, and so on. Any other configuration is likely to cause confusion and, if used, will have to be well documented.

This flexibility, however, allows you to "correct" the arguments for the hooks. For example, L<sup>A</sup>T<sub>E</sub>X's `\refstepcounter` has a single argument, the name of the counter. The `cleveref` package adds an optional argument to `\refstepcounter`, making the name of the counter argument `#2`. If the author of `cleveref` wanted, for whatever reason, to add hooks to `\refstepcounter`, to preserve compatibility he could write something along the lines of:

```
\NewHookWithArguments{cmd/refstepcounter/before}{1}
\renewcommand\refstepcounter[2] [〈default〉]{%
 \UseHookWithArguments{cmd/refstepcounter/before}{1}{#2}%
 <code for \refstepcounter>}
```

so that the mandatory argument, which is arg `#2` in the definition, would still be seen as `#1` in the hook code.

Another possibility would be to place the optional argument as the second argument for the hook, so that people looking for it would be able to use it. In either case, it would have to be well documented to cause as little confusion as possible.

## 4 The Implementation

### 4.1 Execution plan

To add `before` and `after` hooks to a command we will need to peek into the definition of a command, which is always a tricky thing to do. Some cases are easy because we

know how the command was defined, so we can assume how its `\parameter` looks like (for example a command defined with `\newcommand` may have an optional argument followed by a run of mandatory arguments), so we can just expand that command and make it grab #1, #2, etc. as arguments and define it all back with the hooks added.

Life's usually not that easy, so with some commands we can't do that (a #1 might as well be `#1212` instead of the expected `#612`, for example) so we need to resort to "patching" the command: read its `\meaning`, and tokenize it again with `\scantokens` and hope for the best.

So the overall plan is:

1. Check if a command is of a known type (that is, defined with `\newcommand`<sup>20</sup>, `\DeclareRobustCommand`, or `\New(Expandable)DocumentCommand`), and if is, take appropriate action.
2. If the command is not a known type, we'll check if the command can be patched. Two things will prevent a command from being patched: if it was defined in a nonstandard catcode setting, or if it is an internal expl3 command with `_⟨module⟩` in its name, in which case we refuse to patch.
3. If the command was defined in nonstandard catcode settings, we will try a few standard ones to try our best to carry out the patching. If this doesn't help either, the code will give up and throw an error.

```

1 ⟨@=hook⟩
2 ⟨*2ekernel | latexrelease⟩
3 \ExplSyntaxOn
4 ⟨latexrelease⟩ \NewModuleRelease{2021/06/01}{ltcmdhooks}
5 ⟨latexrelease⟩ {The~hook~management~system~for~commands}

```

## 4.2 Variables

Pairs of `\if<cmd>.. \patch<cmd>` to be used with `\robust@command@act` when looking for a known patching rule. This token list is exposed because we see some future applications (with very specialized packages, such as `etoolbox` that may want to extend the pairs processed. It is not meant for general use which is why it is not documented in the interface documentation above.

```
6 \tl_new:N \g_hook_patch_action_list_tl
```

(End of definition for `\g_hook_patch_action_list_tl`.)

`\l__hook_patch_num_args_int` The number of arguments in a macro being patched.

```
7 \int_new:N \l__hook_patch_num_args_int
```

(End of definition for `\l__hook_patch_num_args_int`.)

`\l__hook_patch_prefixes_tl` The prefixes and parameters of the definition for the macro being patched.

```

8 \tl_new:N \l__hook_patch_prefixes_tl
9 \tl_new:N \l__hook_param_text_tl
10 \tl_new:N \l__hook_replace_text_tl
```

---

<sup>20</sup>It's not always possible to reliably detect this case because a command defined with no optional argument is indistinguishable from a `\def`ed command.

|                              |                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------|
|                              | (End of definition for \l_hook_patch_prefixes_t1, \l_hook_param_text_t1, and \l_hook_replace_text_t1.)                |
| \c_hook_hash_t1              | Two constant token lists that contain one and two parameter tokens.                                                   |
| \c_hook_hashes_t1            | <pre> 11 \tl_const:Nn \c_hook_hash_t1 { # } 12 \tl_const:Nn \c_hook_hashes_t1 { # # }</pre>                           |
|                              | (End of definition for \c_hook_hash_t1 and \c_hook_hashes_t1.)                                                        |
| \_hook_exp_not:NN            | Two temporary macros that change depending on the macro being patched.                                                |
| \_hook_def_cmd:w             | <pre> 13 \cs_new_eq:NN \_hook_exp_not:NN ? 14 \cs_new_eq:NN \_hook_def_cmd:w ?</pre>                                  |
|                              | (End of definition for \_hook_exp_not:NN and \_hook_def_cmd:w.)                                                       |
| \q_hook_recursion_tail       | Internal quarks for recursion: they can't appear in any macro being patched.                                          |
| \q_hook_recursion_stop       | <pre> 15 \quark_new:N \q_hook_recursion_tail 16 \quark_new:N \q_hook_recursion_stop</pre>                             |
|                              | (End of definition for \q_hook_recursion_tail and \q_hook_recursion_stop.)                                            |
| \g_hook_delayed_patches_prop | A list containing the patches delayed to \begin{document}, so that patching is not attempted twice.                   |
|                              | <pre> 17 \prop_new:N \g_hook_delayed_patches_prop</pre>                                                               |
|                              | (End of definition for \g_hook_delayed_patches_prop.)                                                                 |
| \_hook_patch_debug:x         | A helper for patching debug info.                                                                                     |
|                              | <pre> 18 \cs_new_protected:Npn \_hook_patch_debug:x #1 19   { \_hook_debug:n { \iow_term:x { [lthooks]~#1 } } }</pre> |
|                              | (End of definition for \_hook_patch_debug:x.)                                                                         |

### 4.3 Variants

|               |                                                             |
|---------------|-------------------------------------------------------------|
| \tl_rescan:nV | expl3 function variants used throughout the code.           |
|               | <pre> 20 \cs_generate_variant:Nn \tl_rescan:nn { nV }</pre> |
|               | (End of definition for \tl_rescan:nV.)                      |

### 4.4 Patching or delaying

Before \begin{document} all patching is delayed.

|                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \_hook_try_put_cmd_hook:n<br>\_hook_try_put_cmd_hook:w | This function is called from within \AddToHook, when code is first added to a generic cmd hook. If it is called within the preamble, it delays the action until \begin{document}; otherwise it tries to update the hook.                                                                                                                                                                                                                                                  |
|                                                        | <pre> 21 ⟨latexrelease⟩\IncludeInRelease{2021/11/15}{\_hook_try_put_cmd_hook:n}% 22 ⟨latexrelease⟩                                {Standardise-generic-hook-names} 23 \cs_new_protected:Npn \_hook_try_put_cmd_hook:n #1 24   { \_hook_try_put_cmd_hook:w #1 / / \s_hook_mark {#1} } 25 \cs_new_protected:Npn \_hook_try_put_cmd_hook:w 26   #1 / #2 / #3 / #4 \s_hook_mark #5 27   { 28     \_hook_debug:n { \iow_term:n { -&gt;~Adding~cmd~hook~to~'#2'~(#3): } }</pre> |

```

29 \exp_args:Nc __hook_patch_cmd_or_delay:Nnn {#2} {#2} {#3}
30 }
31 \end{IncludeInRelease}
32 \IncludeInRelease{2021/06/01}{__hook_try_put_cmd_hook:n}%
33 {\Standardise-generic-hook-names}
34 \cs_new_protected:Npn __hook_try_put_cmd_hook:n #1
35 { __hook_try_put_cmd_hook:w #1 / / / \s__hook_mark {#1} }
36 \cs_new_protected:Npn __hook_try_put_cmd_hook:w
37 #1 / #2 / #3 / #4 \s__hook_mark #5
38 \cs_new_protected:Npn __hook_debug:n { \iow_term:n { ->-Adding-cmd-hook-to-'#2'-'#3': } }
39 \str_case:nnTF {#3}
40 { { before } { } { after } { } }
41 { \exp_args:Nc __hook_patch_cmd_or_delay:Nnn {#2} {#2} {#3} }
42 { \msg_error:nnnn { hooks } { wrong-cmd-hook } {#2} {#3} }
43 \end{IncludeInRelease}
44 \end{IncludeInRelease}
45 \end{IncludeInRelease}

```

(End of definition for `\__hook_try_put_cmd_hook:n` and `\__hook_try_put_cmd_hook:w`.)

`\__hook_patch_cmd_or_delay:Nnn \__hook_cmd_begindocument_code:` In the preamble, `\__hook_patch_cmd_or_delay:Nnn` just adds the patch instruction to a property list to be executed later.

```

46 \cs_new_protected:Npn __hook_patch_cmd_or_delay:Nnn #1 #2 #3
47 {
48 __hook_debug:n { \iow_term:n { ->-Add-generic-cmd-hook-for-'#2-'#3'. } }
49 __hook_debug:n
50 { \iow_term:n { !~In-the-preamble:-delaying. } }
51 \prop_gput:Nnn \g__hook_delayed_patches_prop { #2 / #3 }
52 { __hook_cmd_try_patch:nn {#2} {#3} }
53 }

```

The delayed patches are added to a property list to prevent duplication, and the code stored in the property list for each key is executed. The function `\__hook_patch_cmd_or_delay:Nnn` is also redefined to be `\__hook_patch_command:Nnn` so that no further delaying is attempted.

```

54 \cs_new_protected:Npn __hook_cmd_begindocument_code:
55 {
56 \cs_gset_eq:NN __hook_patch_cmd_or_delay:Nnn __hook_patch_command:Nnn
57 \prop_map_function:NN \g__hook_delayed_patches_prop { \use_i:nn }
58 \prop_gclear:N \g__hook_delayed_patches_prop
59 \cs_undefine:N __hook_cmd_begindocument_code:
60 }
61 \g@addto@macro \kernel@after@begindocument
62 { __hook_cmd_begindocument_code: }

```

(End of definition for `\__hook_patch_cmd_or_delay:Nnn` and `\__hook_cmd_begindocument_code`.)

`\__hook_cmd_try_patch:nn` At `\begin{document}` tries patching the command if the hook was not manually created in the meantime. If the document does not exist, no error is raised here as it may hook into a package that wasn't loaded. Hooks added to commands in the document body still raise an error if the command is not defined.

```

63 \cs_new_protected:Npn __hook_cmd_try_patch:nn #1 #2
64 {
65 __hook_debug:n

```

```

66 { \iow_term:x { ->~\string\begin{document}~try~cmd / #1 / #2. } }
67 __hook_if_declared:nTF { cmd / #1 / #2 }
68 {
69 __hook_debug:n
70 { \iow_term:n { .->~Giving~up~hook~already~created. } }
71 }
72 {
73 \cs_if_exist:cT {#1}
74 { \exp_args:Nc __hook_patch_command:Nnn {#1} {#1} {#2} }
75 }
76 }

```

(End of definition for `\__hook_cmd_try_patch:nn`.)

## 4.5 Patching commands

```

__hook_patch_command:Nnn
__hook_patch_check>NNnn
__hook_if_public_command:NTF
__hook_if_public_command:w

```

`\__hook_patch_command:Nnn` will do some sanity checks on the argument to detect if it is possible to add hooks to the command, and raises an error otherwise. If the command can contain hooks, then it uses `\robust@command@act` to find out what type is the command, and patch it accordingly.

```

77 \cs_new_protected:Npn __hook_patch_command:Nnn #1 #2 #3
78 {
79 __hook_patch_debug:x { analyzing~'\token_to_str:N #1' }
80 __hook_patch_debug:x { \token_to_str:N #1 = \token_to_meaning:N #1 }
81 __hook_patch_check>NNnn \cs_if_exist:NTF #1 { undef }
82 {
83 __hook_patch_debug:x { +--control~sequence~is~defined }
84 __hook_patch_check>NNnn \token_if_macro:NTF #1 { macro }
85 {
86 __hook_patch_debug:x { +--control~sequence~is~a~macro }
87 __hook_patch_check>NNnn __hook_if_public_command:NTF #1 { expl3 }
88 {
89 __hook_patch_debug:x { +--macro~is~not~private }
90 \robust@command@act
91 \g_hook_patch_action_list_tl #1
92 __hook_retokenize_patch:Nnn { #1 {#2} {#3} }
93 }
94 }
95 }
96 }

```

And here's the auxiliary used above:

```

97 \cs_new_protected:Npn __hook_patch_check>NNnn #1 #2 #3 #4
98 {
99 #1 #2 {#4}
100 {
101 \msg_error:nnxx { hooks } { cant~patch }
102 { \token_to_str:N #2 } {#3}
103 }
104 }

```

and a conditional `\__hook_if_public_command:NTF` to check if a command has `_` in its name (no other checking is performed). Primitives with :D in their name could be included here, but they are already discarded in the `\token_if_macro:NTF` test above.

```

105 \use:x
106 {
107 \prg_new_protected_conditional:Nnn
108 \exp_not:N __hook_if_public_command:N ##1 { TF }
109 {
110 \exp_not:N \exp_last_unbraced:Nf
111 \exp_not:N __hook_if_public_command:w
112 { \exp_not:N \cs_to_str:N ##1 }
113 \tl_to_str:n { _ } \s_hook_mark
114 }
115 }
116 \exp_last_unbraced:NNNNo
117 \cs_new_protected:Npn __hook_if_public_command:w
118 #1 \tl_to_str:n { _ } #2 \s_hook_mark
119 {
120 \tl_if_empty:nTF {#2}
121 { \prg_return_true: }
122 { \prg_return_false: }
123 }

```

(End of definition for `\__hook_patch_command:Nnn` and others.)

#### 4.5.1 Patching by expansion and redefinition

This is the list of known command types and the function that patches the command hooks into them. The conditionals are taken from `\ShowCommand`, `\NewCommandCopy` and `\__kernel_cmd_if_xparse:NTF` defined in `ltcmd`.

```

124 \tl_gset:Nn \g_hook_patch_action_list_tl
125 {
126 { \c_if@DeclareRobustCommand __hook_patch_DeclareRobustCommand:Nnn }
127 { \c_if@newcommand __hook_patch_newcommand:Nnn }
128 { __kernel_cmd_if_xparse:NTF __hook_cmd_patch_xparse:Nnn }
129 }

```

(End of definition for `\g_hook_patch_action_list_tl`.)

At this point we know that the commands can be patched by expanding then redefining. These are the cases of commands defined with `\newcommand` with an optional argument or with `\DeclareRobustCommand`.

With `\__hook_patch_DeclareRobustCommand:Nnn` we check if the command has an optional argument (with a test counter-intuitively called `\c_if@newcommand`; also make sure the command doesn't take args by calling `\robust@command@chk@safe`). If so, we pass the patching action to `\__hook_patch_newcommand:Nnn`, otherwise we call the patching engine `\__hook_patch_expand_redefine:NNnn` with a `\c_false_bool` to indicate that there is no optional argument.

```

130 \cs_new_protected:Npn __hook_patch_DeclareRobustCommand:Nnn #1
131 {
132 \exp_args:Nc __hook_patch_DeclareRobustCommand_aux:Nnn
133 { \cs_to_str:N #1 ~ }
134 }
135 \cs_new_protected:Npn __hook_patch_DeclareRobustCommand_aux:Nnn #1
136 {
137 \robust@command@chk@safe #1

```

```

138 { \@if@newcommand #1 }
139 { \use_ii:nn }
140 { __hook_patch_newcommand:Nnn }
141 { __hook_patch_expand_redefine:NNnn \c_false_bool }
142 #1
143 }

```

(End of definition for `\__hook_patch_DeclareRobustCommand:Nnn`.)

`\__hook_patch_newcommand:Nnn` If the command was defined with `\newcommand` and an optional argument, call the patching engine with a `\c_true_bool` to flag the presence of an optional argument, and with `\\command` to patch the actual code for `\command`.

```

144 \cs_new_protected:Npn __hook_patch_newcommand:Nnn #1
145 {
146 \exp_args:NNc __hook_patch_expand_redefine:NNnn \c_true_bool
147 { \c_backslash_str \cs_to_str:N #1 }
148 }

```

(End of definition for `\__hook_patch_newcommand:Nnn`.)

`\__hook_cmd_patch_xparse:Nnn` And for commands defined by the `xparse` commands use this for patching:

```

149 \cs_new_protected:Npn __hook_cmd_patch_xparse:Nnn #1
150 {
151 \exp_args:NNc __hook_patch_expand_redefine:NNnn \c_false_bool
152 { \cs_to_str:N #1 ~ code }
153 }

```

(End of definition for `\__hook_cmd_patch_xparse:Nnn`.)

`\__hook_patch_expand_redefine:NNnn`  
`\__hook_redefine_with_hooks:Nnn`  
`\__hook_make_prefixes:w` Now the real action begins. Here we have in `#1` a boolean indicating if the command has a leading [...] delimited argument, in `#2` the command control sequence, in `#3` the name of the command (note that `#1 ≠ \csname#2\endcsname` at this point!), and in `#4` the hook position, either `before` or `after`.

Patching with expansion+redefinition is trickier than it looks like at first glance. Suppose the simple definition:

```
\def\foo#1{#1##2}
```

When defined, its `\replacement_text` will be a token list containing:

```
out_param 1, mac_param #, character 2
```

Then, after expanding `\foo{##1}` (here `##` denotes a single `#`) we end up with a token list with `out_param 1` replaced:

```
mac_param #, character 1, mac_param #, character 2
```

that is, the definition would be:

```
\def\foo#1{#1#2}
```

which obviously fails, because the original input in the definition was `##` but TeX reduced that to a single parameter token `#` when carrying out the definition. That leaves no room for a clever solution with (say) `\unexpanded`, because anything that would double the second `#`, would also (incorrectly) double the first, so there's not much to do other than a manual solution.

There are three cases we can distinguish to make things hopefully faster on simpler cases:

1. a macro with no parameters;
2. a macro with no parameter tokens in its definition;
3. a macro with parameters *and* parameter tokens.

The first case is trivial: if the macro has no parameters, we can just use `\unexpanded` around it, and if there is a parameter token in it, it is handled correctly (the macro can be treated as a `t1` variable).

The second case requires looking at the `<replacement text>` of the macro to see if it has a parameter token in there. If it does not, then there is no worry, and the macro can be redefined normally (without `\unexpanded`).

The third case, as usual, is the devious one. Here we'll have to loop through the definition token by token, and double every parameter token, so that this case can be handled like the previous one.

```

154 <latexrelease> \IncludeInRelease{2023/06/01}{_hook_patch_expand_redefine:NNnn}
155 <latexrelease> {cmd~hooks~with~args}
156 \cs_new_protected:Npn _hook_patch_expand_redefine:NNnn #1 #2 #3 #4
157 {
158 _hook_patch_debug:x { ++~command~can~be~patched~without~rescanning }

```

We'll start by counting the number of arguments in the command by counting the number of characters in the `\cs_parameter_spec:N` of the macro, divided by two, and subtracting one if the command has an optional argument (that is, an extra `[]` in its `<parameter text>`).

```

159 \int_set:Nn \l__hook_patch_num_args_int
160 {
161 \exp_args:Nf \str_count:n { _kernel_cs_parameter_spec:N #2 } / 2
162 \bool_if:NT #1 { -1 }
163 }

```

Now build two token lists:

`\l__hook_param_text_t1` will contain the `<parameter text>` to be used when redefining the macro. It should be identical to the `<parameter text>` used when originally defining that macro.

`\l__hook_replace_text_t1` will contain braced pairs of `\c__hook_hashes_t1<num>` to feed to the macro when expanded. This token list as well as the previous will have the first item surrounded by `[...]` in the case of an optional argument.

The use of `\c__hook_hashes_t1` here is to differentiate actual parameters in the macro from parameter tokens in the original definition of the macro. Later on, `\c__hook_hashes_t1` is either replaced by actual parameter tokens, or expanded into them.

```

164 \int_compare:nNnTF { \l__hook_patch_num_args_int } > { \c_zero_int }
165 {

```

We'll first check if the command has any parameter token in its definition (feeding it empty arguments), and set `\_hook_exp_not:n` accordingly. `\_hook_exp_not:n` will be used later to either leave `\c__hook_hashes_t1` or expand it, and also to remember the result of `\_hook_if_has_hash:nTF` to avoid testing twice (the test can be rather slow).

```

166 \tl_set:Nx \l__hook_tmpa_t1 { \bool_if:NTF #1 { [] } { { } } }
167 \int_step_inline:nnn { 2 } { \l__hook_patch_num_args_int }

```

```

168 { \tl_put_right:Nn \l__hook_tmpa_tl { { } } }
169 \exp_args:NNo \exp_args:No __hook_if_has_hash:nTF
170 { \exp_after:wN #2 \l__hook_tmpa_tl }
171 { \cs_set_eq:NN __hook_exp_not:n \exp_not:n }
172 { \cs_set_eq:NN __hook_exp_not:n \use:n }
173 \cs_set_protected:Npn __hook_tmp:w ##1 ##2
174 {
175 ##1 \l__hook_param_text_tl { \use:n ##2 }
176 ##1 \l__hook_replace_text_tl { __hook_exp_not:n {##2} }
177 }

```

Here we'll conditionally add [...] around the first parameter:

```

178 \bool_if:NTF #1
179 { __hook_tmp:w \tl_set:Nx { [\c__hook_hashes_tl 1] } }
180 { __hook_tmp:w \tl_set:Nx { { \c__hook_hashes_tl 1 } } }

```

Then, for every parameter from the second, just add it normally:

```

181 \int_step_inline:nnn { 2 } { \l__hook_patch_num_args_int }
182 { __hook_tmp:w \tl_put_right:Nx { { \c__hook_hashes_tl ##1 } } }

```

Now, if the command has any parameter token in its definition (then `\__hook_exp_not:n` is `\exp_not:n`), call `\__hook_double_hashes:n` to double them, and replace every `\c__hook_hashes_tl` by #:

```

183 \tl_set:Nx \l__hook_replace_text_tl
184 { \exp_not:N #2 \exp_not:V \l__hook_replace_text_tl }
185 \tl_set:Nx \l__hook_replace_text_tl
186 {
187 \token_if_eq_meaning:NNTF __hook_exp_not:n \exp_not:n
188 { \exp_args:NNV \exp_args:No __hook_double_hashes:n }
189 { \exp_args:NV \exp_not:o }
190 \l__hook_replace_text_tl
191 }

```

And now, set a few auxiliaries for the case that the macro has parameters, so it won't be passed through `\unexpanded` (twice):

```

192 \cs_set_eq:NN __hook_def_cmd:w \tex_gdef:D
193 \cs_set_eq:NN __hook_exp_not:NN \prg_do_nothing:
194 }
195 {

```

In the case the macro has no parameters, we'll treat it as a token list and things are much simpler (expansion control looks a bit complicated, but it's just a pair of `\exp_not:N` preventing another `\exp_not:n` from expanding):

```

196 \tl_clear:N \l__hook_param_text_tl
197 \tl_set_eq:NN \l__hook_replace_text_tl #2
198 \cs_set_eq:NN __hook_def_cmd:w \tex_xdef:D
199 \cs_set:Npn __hook_exp_not:NN ##1 { \exp_not:N ##1 \exp_not:N }
200 }

```

Before redefining, we need to also get the prefixes used when defining the command. Here we ensure that the `\escapechar` is printable, otherwise a macro defined with prefixes `\protected \long` will have it `\meaning` printed as `protectedlong`, making life unnecessarily complicated. Here the `\escapechar` is changed to /, then we loop between pairs of /.../ extracting the prefixes.

```

201 \group_begin:
202 \int_set:Nn \tex_escapechar:D { '\! }

```

```

203 \use:x
204 {
205 \group_end:
206 \tl_set:Nx \exp_not:N \l__hook_patch_prefixes_tl
207 { \exp_not:N _hook_make_prefixes:w \cs_prefix_spec:N #2 / / }
208 }

```

Here we redefine the hook to have the right number of arguments. Disabling the hook, undefining the `parameter` token list then calling `\_hook_make_usable:nn` are enough to redefine the hook to the extent we want. Code stored in the hook and other metadata about it are not lost in the process.

```

209 _hook_disable:n { cmd / #3 / #4 }
210 \cs_undefine:c { c__hook_cmd / #3 / #4_parameter_tl }
211 _hook_make_usable:nn { cmd / #3 / #4 } { \l__hook_patch_num_args_int }

```

Now call `\_hook_redefine_with_hooks:Nnnn` with the macro being redefined in #1, then `\UseHook{cmd/<name>/before}` in #2 or `\UseHook{cmd/<name>/after}` in #3 (one is always empty), and in #4 the `<replacement text>` of the macro.

```

212 \use:e
213 {
214 _hook_redefine_with_hooks:Nnnn \exp_not:N #2
215 \str_if_eq:nnTF {#4} { after }
216 { \use_i:i:nn }
217 { \use:nn }
218 {
219 _hook_exp_not>NN \exp_not:N \UseHookWithArguments
220 { cmd / #3 / #4 } { \int_use:N \l__hook_patch_num_args_int }
221 _hook_braced_parameter:n { cmd / #3 / #4 }
222 }
223 { { } }
224 { _hook_exp_not>NN \exp_not:V \l__hook_replace_text_tl }
225 }

```

Finally, update the hook code.

```

226 _hook_update_hook_code:n { cmd / #3 / #4 }
227 }
228 \EndIncludeInRelease
229 \IncludeInRelease{2021/06/01}{_hook_patch_expand_redefine:Nnnn}
230 {cmd~hooks~with~args}
231 \cs_gset_protected:Npn _hook_patch_expand_redefine:NNnn #1 #2 #3 #4
232 {
233 _hook_patch_debug:x { ++-command-can-be-patched-without-rescanning }
234 \int_set:Nn \l__hook_patch_num_args_int
235 {
236 \exp_args:Nf \str_count:n { \kernel_cs_parameter_spec:N #2 } / 2
237 \bool_if:NT #1 { -1 }
238 }
239 \int_compare:nNnTF { \l__hook_patch_num_args_int } > { \c_zero_int }
240 {
241 \tl_set:Nx \l__hook_tmptl { \bool_if:NTF #1 { [] } { { } } }
242 \int_step_inline:nnn { 2 } { \l__hook_patch_num_args_int }
243 { \tl_put_right:Nn \l__hook_tmptl { { } } }
244 \exp_args:NNo \exp_args:No _hook_if_has_hash:nTF
245 { \exp_after:WN #2 \l__hook_tmptl }
246 { \cs_set_eq:NN _hook_exp_not:n \exp_not:n }

```

```

247 〈latexrelease〉 { \cs_set_eq:NN __hook_exp_not:n \use:n }
248 〈latexrelease〉 \cs_set_protected:Npn __hook_tmp:w ##1 ##2
249 〈latexrelease〉 {
250 〈latexrelease〉 ##1 \l__hook_param_text_tl { \use:n ##2 }
251 〈latexrelease〉 ##1 \l__hook_replace_text_tl { __hook_exp_not:n {##2} }
252 〈latexrelease〉 }
253 〈latexrelease〉 \bool_if:NTF #1
254 〈latexrelease〉 { __hook_tmp:w \tl_set:Nx { [\c__hook_hash_tl 1] } }
255 〈latexrelease〉 { __hook_tmp:w \tl_set:Nx { { \c__hook_hash_tl 1 } } }
256 〈latexrelease〉 \int_step_inline:nnn { 2 } { \l__hook_patch_num_args_int }
257 〈latexrelease〉 { __hook_tmp:w \tl_put_right:Nx { { \c__hook_hash_tl ##1 } } }
258 〈latexrelease〉 \tl_set:Nx \l__hook_replace_text_tl
259 〈latexrelease〉 { \exp_not:N #2 \exp_not:V \l__hook_replace_text_tl }
260 〈latexrelease〉 \tl_set:Nx \l__hook_replace_text_tl
261 〈latexrelease〉 {
262 〈latexrelease〉 \token_if_eq_meaning:NNTF __hook_exp_not:n \exp_not:n
263 〈latexrelease〉 { \exp_args:NNV \exp_args:No __hook_double_hashes:n }
264 〈latexrelease〉 { \exp_args:NV \exp_not:o }
265 〈latexrelease〉 \l__hook_replace_text_tl
266 〈latexrelease〉 }
267 〈latexrelease〉 \cs_set_eq:NN __hook_def_cmd:w \tex_gdef:D
268 〈latexrelease〉 \cs_set_eq:NN __hook_exp_not:NN \prg_do_nothing:
269 〈latexrelease〉 }
270 〈latexrelease〉 {
271 〈latexrelease〉 \tl_clear:N \l__hook_param_text_tl
272 〈latexrelease〉 \tl_set_eq:NN \l__hook_replace_text_tl #2
273 〈latexrelease〉 \cs_set_eq:NN __hook_def_cmd:w \tex_xdef:D
274 〈latexrelease〉 \cs_set:Npn __hook_exp_not:NN ##1 { \exp_not:N ##1 \exp_not:N }
275 〈latexrelease〉 }
276 〈latexrelease〉 \group_begin:
277 〈latexrelease〉 \int_set:Nn \tex_escapechar:D { ‘\` }
278 〈latexrelease〉 \use:x
279 〈latexrelease〉 {
280 〈latexrelease〉 \group_end:
281 〈latexrelease〉 \tl_set:Nx \exp_not:N \l__hook_patch_prefixes_tl
282 〈latexrelease〉 { \exp_not:N __hook_make_prefixes:w \cs_prefix_spec:N #2 / / }
283 〈latexrelease〉 }
284 〈latexrelease〉 \use:x
285 〈latexrelease〉 {
286 〈latexrelease〉 __hook redefine_with_hooks:Nnnn \exp_not:N #2
287 〈latexrelease〉 \str_if_eq:nnTF {#4} { after }
288 〈latexrelease〉 { \use_i:i:nn }
289 〈latexrelease〉 { \use:nn }
290 〈latexrelease〉 { { __hook_exp_not:NN \exp_not:N \UseHook { cmd / #3 / #4 } } }
291 〈latexrelease〉 { { } }
292 〈latexrelease〉 { __hook_exp_not:NN \exp_not:V \l__hook_replace_text_tl }
293 〈latexrelease〉 }
294 〈latexrelease〉 }
295 〈latexrelease〉 \EndIncludeInRelease

```

Now that all the needed tools are ready, without further ado we'll redefine the command. The definition uses the prefixes gathered in `\l__hook_patch_prefixes_tl`, a primitive `\__hook_def_cmd:w` (which is `\tex_gdef:D` or `\tex_xdef:D`) to avoid adding extra prefixes, and the `parameter text` from `\l__hook_param_text_tl`.

Then finally, in the body of the definition, we insert #2, which is `cmd/#1/before` or empty, #4 which is the `<replacement text>`, and #3 which is `cmd/#1/after` or empty.

```

296 \cs_new_protected:Npn __hook_redefine_with_hooks:Nnnn #1 #2 #3 #4
297 {
298 \l__hook_patch_prefixes_tl
299 \exp_after:wN __hook_def_cmd:w
300 \exp_after:wN #1 \l__hook_param_text_tl
301 { #2 #4 #3 }
302 }
```

Here's the auxiliary that makes the prefix control sequences for the redefinition. Each item has to be `\tl_trim_spaces:n`'d because the last item (and not any other) has a trailing space.

```

303 \cs_new:Npn __hook_make_prefixes:w / #1 /
304 {
305 \tl_if_empty:nF {#1}
306 {
307 \exp_not:c { tex_ \tl_trim_spaces:n {#1} :D }
308 __hook_make_prefixes:w /
309 }
310 }
```

(End of definition for `\__hook_patch_expand_redefine:NNnn`, `\__hook_redefine_with_hooks:Nnnn`, and `\__hook_make_prefixes:w`.)

Here are some auxiliaries for the contraption above.

`\__hook_if_has_hash:p:n` `\__hook_if_has_hash:nTF` searches the token list #1 for a catcode 6 token, and if any is found, it returns `true`, and `false` otherwise. The searching doesn't care about preserving groups or spaces: we can ignore those safely (braces are removed) so that searching is as fast as possible.

```

311 \prg_new_conditional:Npnn __hook_if_has_hash:n #1 { TF }
312 { __hook_if_has_hash:w #1 ## \s__hook_mark }
313 \cs_new:Npn __hook_if_has_hash:w #1
314 {
315 \tl_if_single_token:nTF {#1}
316 {
317 \token_if_eq_catcode:NNTF ## #1
318 { __hook_if_has_hash_check:w }
319 { __hook_if_has_hash:w }
320 }
321 { __hook_if_has_hash:w #1 }
322 }
323 \cs_new:Npn __hook_if_has_hash_check:w #1 \s__hook_mark
324 { \tl_if_empty:nTF {#1} { \prg_return_false: } { \prg_return_true: } }
```

(End of definition for `\__hook_if_has_hash:nTF`, `\__hook_if_has_hash:w`, and `\__hook_if_has_hash_check:w`.)

`\__hook_double_hashes:n` loops through the token list #1 and duplicates any catcode 6 token, and expands tokens `\ifx-equal` to `\c__hook_hashes_tl`, and leaves all other tokens `\notexpanded` with `\exp_not:N`. Unfortunately pairs of explicit catcode 1 and catcode 2 character tokens are normalised to `{1}` and `}1` because it's not feasible to expandably detect the character code (*maybe* it could be done using something along the

lines of <https://tex.stackexchange.com/a/527538>, but it's far too much work for close to zero benefit).

`\_\_hook\_double\_hashes:w` is the tail-recursive loop macro, that tests which of the three types of item is in the head of the token list.

```

325 \cs_new:Npn __hook_double_hashes:n #1
326 { __hook_double_hashes:w #1 \q__hook_recursion_tail \q__hook_recursion_stop }
327 \cs_new:Npn __hook_double_hashes:w #1 \q__hook_recursion_stop
328 {
329 \tl_if_head_is_N_type:nTF {#1}
330 { __hook_double_hashes_output:N }
331 {
332 \tl_if_head_is_group:nTF {#1}
333 { __hook_double_hashes_group:n }
334 { __hook_double_hashes_space:w }
335 }
336 #1 \q__hook_recursion_stop
337 }

__hook_double_hashes_output:N checks for the end of the token list, then checks if the token is \c__hook_hashes_tl, and if so just leaves it.

338 \cs_new:Npn __hook_double_hashes_output:N #1
339 {
340 \if_meaning:w \q__hook_recursion_tail #1
341 __hook_double_hashes_stop:w
342 \fi:
343 \if:w ?
344 \if_meaning:w \c__hook_hash_tl #1 ! \fi:
345 \if_meaning:w \c__hook_hashes_tl #1 ! \fi:
346 ?
347 \else:

```

(this `\use_i:nnnn` uses `\fi:` and consumes `\use:n`, the whole `\if_catcode:w` block, and the `\exp_not:N`, leaving just `#1` which is `\c_\_hook_hashes_tl`.)

```

348 \use_i:nnnn
349 \fi:
350 \use:n
351 {

```

If `#1` is not `\c_\_hook_hashes_tl`, then check if its catcode is 6, and if so, leave it doubled in `\exp_not:n` and consume the following `\exp_not:N #1`.

```

352 \if_catcode:w ## \exp_not:N #1
353 \exp_after:wN \use_ii:nnnn
354 \fi:
355 \use_none:n
356 { \exp_not:n { #1 #1 } }
357 }

```

If both previous tests returned `false`, then leave the token unexpanded and resume the loop.

```

358 \exp_not:N #1
359 __hook_double_hashes:w
360 }
361 \cs_new:Npn __hook_double_hashes_stop:w #1 \q__hook_recursion_stop { \fi: }

```

Dealing with spaces and grouped tokens is trivial:

```

362 \cs_new:Npn __hook_double_hashes_group:n #1
363 { { __hook_double_hashes:n {#1} } __hook_double_hashes:w }
364 \exp_last_unbraced:NNo
365 \cs_new:Npn __hook_double_hashes_space:w \c_space_tl
366 { ~ __hook_double_hashes:w }
```

(End of definition for `\__hook_double_hashes:n` and others.)

#### 4.5.2 Patching by retokenization

At this point we've drained the possibilities of patching a command by expansion-and-redefinition, so we have to resort to patching by retokenizing the command. Patching by retokenization is done by getting the `\meaning` of the command, doing the necessary manipulations on the generated string, and the retokenizing that again by using `\scantokens`.

Patching by retokenization is definitely a riskier business, because it relies that the tokens printed by `\meaning` produce the exact same tokens as the ones in the original definition. That is, the catcode régime must be exactly(ish) the same, and there is no way of telling except by trial and error.

`\__hook_retokenize_patch:Nnn`

This is the macro that will control the whole process. First we'll try out one final, rather trivial case, of a command with no arguments; that is, a token list. This case can be patched with the expand-and-redefine routine but it has to be the very last case tested for, because most (all?) robust commands start with a top-level macro with no arguments, so testing this first would short-circuit `\robust@command@act` and the top-level macros would be incorrectly patched. In that case, we just check if the `\cs_parameter_spec:N` is empty, and call `\__hook_patch_expand_redefine>NNnn`.

```

367 \cs_new_protected:Npn __hook_retokenize_patch:Nnn #1 #2 #3
368 {
369 \str_if_eq:eeTF { __kernel_cs_parameter_spec:N #1 } { }
370 { __hook_patch_expand_redefine:NNnn \c_false_bool #1 {#2} {#3} }
371 {
372 __hook_patch_debug:x { ..~command~can~only~be~patched~by~rescanning }
```

Otherwise, we start the actual patching by retokenization job. The code calls `\__hook_try_patch_with_catcodes:Nnnnw` with a different catcode setting:

- The current catcode setting;
- Switching the catcode of `\C{0}`;
- Switching the `\expl3` syntax on or off;
- Both of the above.

If patching succeeds, `\__hook_try_patch_with_catcodes:Nnnnw` has the side-effect of patching the macro `#1` (which may be an internal from the command whose name is `#2`).

```

373 \tl_set:Nx \l__hook_tmpa_tl
374 {
375 \int_compare:nNnTF { \char_value_catcode:n {`\C{0}} } = { 12 }
376 { \exp_not:N \makeatletter } { \exp_not:N \makeatother }
377 }
378 \tl_set:Nx \l__hook_tmpb_tl
```

```

379 {
380 \bool_if:NTF \l__kernel_expl_bool
381 { \ExplSyntaxOff } { \ExplSyntaxOn }
382 }
383 \use:x
384 {
385 \exp_not:N _hook_try_patch_with_catcodes:Nnnnw
386 \exp_not:n { #1 {#2} {#3} }
387 { \prg_do_nothing: }
388 { \exp_not:V \l__hook_tmpa_tl } % @
389 { \exp_not:V \l__hook_tmpb_tl } % _:
390 {
391 \exp_not:V \l__hook_tmpa_tl % @
392 \exp_not:V \l__hook_tmpb_tl % _:
393 }
394 }
395 \q_recursion_tail \q_recursion_stop

```

If no catcode setting succeeds, give up and raise an error. The command isn't changed in any way in that case.

```

396 {
397 \msg_error:nxxx { hooks } { cant-patch }
398 { \c_backslash_str #2 } { retok }
399 }
400 }
401 }

```

(End of definition for `\_hook_retokenize_patch:Nnn`.)

This function is a simple wrapper around `\_hook_cmd_if_scannable:NnTF` and `\_hook_patch_retokenize:Nnn` if the former returns `\true`, plus some debug messages.

```

402 \latexrelease\IncludeInRelease{2023/06/01}{_hook_try_patch_with_catcodes:Nnnnw}
403 \latexrelease {cmd~hooks~with~args}
404 \cs_new_protected:Npn _hook_try_patch_with_catcodes:Nnnnw #1 #2 #3 #4
405 {
406 \quark_if_recursion_tail_stop_do:nn {#4} { \use:n }
407 _hook_patch_debug:x { ++-trying-to-patch-by-retokenization }
408 _hook_cmd_if_scannable:NnTF {#1} {#4}
409 {
410 _hook_patch_debug:x { +--+macro-can-be-retokenized-cleanly }
411 _hook_patch_debug:x { ==-retokenizing-macro-now }
412 _hook_patch_retokenize:Nnnn #1 { cmd / #2 / #3 } {#3} {#4}
413 \use_i_delimit_by_q_recursion_stop:nw \use_none:n
414 }
415 {
416 _hook_patch_debug:x { ---macro-cannot-be-retokenized-cleanly }
417 _hook_try_patch_with_catcodes:Nnnnw #1 {#2} {#3}
418 }
419 }
420 \latexrelease\EndIncludeInRelease
421 \latexrelease\IncludeInRelease{2021/06/01}{_hook_try_patch_with_catcodes:Nnnnw}
422 \latexrelease {cmd~hooks~with~args}
423 \latexrelease\cs_gset_protected:Npn _hook_try_patch_with_catcodes:Nnnnw #1 #2 #3 #4
424 \latexrelease {
425 \latexrelease \quark_if_recursion_tail_stop_do:nn {#4} { \use:n }

```

```

426 <latexrelease> __hook_patch_debug:x { +--+trying-to-patch-by-retokenization }
427 <latexrelease> __hook_cmd_if_scannable:NnTF {#1} {#4}
428 <latexrelease> {
429 <latexrelease> __hook_patch_debug:x { +--+macro-can-be-retokenized-cleanly }
430 <latexrelease> __hook_patch_debug:x { ==-retokenizing-macro-now }
431 <latexrelease> __hook_patch_retokenize:Nnn #1 {#2} {#3} {#4}
432 <latexrelease> \use_i_delimit_by_q_recursion_stop:nw \use_none:n
433 <latexrelease>
434 <latexrelease> {
435 <latexrelease> __hook_patch_debug:x { --macro-cannot-be-retokenized-cleanly }
436 <latexrelease> __hook_try_patch_with_catcodes:Nnnnw #1 {#2} {#3}
437 <latexrelease>
438 <latexrelease> }
439 <latexrelease>\EndIncludeInRelease

```

(End of definition for \\_\_hook\_try\_patch\_with\_catcodes:Nnnnw.)

\kerneltmpDoNotUse This is an oddity required to be safe (as safe as reasonably possible) when patching the command. The entirety of

```
<prefixes> \def <cs> <parameter text> {<replacement text>}
```

will go through \scantokens. The <parameter text> and <replacement text> are what we are trying to retokenize, so not much worry there. The other items, however, should “just work”, so some care is needed to not use too fancy catcode settings. Therefore we can’t use an expl3-named macro for <cs>, nor the expl3 versions of \def or the <prefixes>. That is why the definitions that will eventually go into \scantokens will use the oddly (but hopefully clearly)-named \kerneltmpDoNotUse:

```
440 \cs_new_eq:NN \kerneltmpDoNotUse !
```

*PhO: Maybe this can be avoided by running the <parameter text> and the <replacement text> separately through \scantokens and then putting everything together at the end.*

(End of definition for \kerneltmpDoNotUse.)

\\_\_hook\_patch\_required\_catcodes: Here are the catcode settings that are *mandatory* when retokenizing commands. These are the minimum necessary settings to perform the definitions: they identify control sequences, which must be escaped with \\_, delimit the definition with \\_1 and \\_2, and mark parameters with \#6. Everything else may be changed, but not these.

```

441 \cs_new_protected:Npn __hook_patch_required_catcodes:
442 {
443 \char_set_catcode_escape:N \\
444 \char_set_catcode_group_begin:N \{
445 \char_set_catcode_group_end:N \}
446 \char_set_catcode_parameter:N \#
447 % \int_set:Nn \tex_endlinechar:D { -1 }
448 % \int_set:Nn \tex_newlinechar:D { -1 }
449 }

```

*PhO: etoolbox sets the \endlinechar and \newlinechar when patching, but as far as I tested these didn’t make much of a difference, so I left them out for now. Maybe \newlinechar=-1 avoids a space token being added after the definition.*

*PhO: If the patching is split by <parameter text> and <replacement text>, then only \# will have to stay in that list.*

*PhO:* Actually now that we patch \UseHook{cmd/foo/before}, all the tokens there need to have the right catcodes, so this list now includes all lowercase letters, U and H, the slash, and whatever characters in the command name... sigh...

(End of definition for `\_hook_patch_required_catcodes:.`)

`\_hook_cmd_if_scannable:NnTF`

Here we'll do a quick test if the command being patched can in fact be retokenized with the specific catcode setting without changing in meaning. The test is straightforward:

1. apply `\meaning` to the command;
2. split the `<prefixes>`, `<parameter text>` and `<replacement text>` and arrange them as

`<prefixes>\def\kerneltmpDoNotUse<parameter text>{\<replacement text>}`

3. rscan that with the given catcode settings, and do the definition; then finally
4. compare `\kerneltmpDoNotUse` with the original command.

If both are `\ifx`-equal, the command can be safely patched.

```

450 \prg_new_protected_conditional:Npnn _hook_cmd_if_scannable:Nn #1 #2 { TF }
451 {
452 \cs_set_eq:NN \kerneltmpDoNotUse \scan_stop:
453 \cs_set_eq:NN _hook_tmp:w \scan_stop:
454 \use:x
455 {
456 \cs_set:Npn _hook_tmp:w
457 #####1 \tl_to_str:n { macro: } #####2 -> #####3 \s__hook_mark
458 { #####1 \def \kerneltmpDoNotUse #####2 {#####3} }
459 \tl_set:Nx \exp_not:N \l__hook_tmpa_tl
460 { \exp_not:N _hook_tmp:w \token_to_meaning:N #1 \s__hook_mark }
461 }
462 \tl_rscan:nV { #2 _hook_patch_required_catcodes: } \l__hook_tmpa_tl
463 \token_if_eq_meaning:NNTF #1 \kerneltmpDoNotUse
464 { \prg_return_true: }
465 { \prg_return_false: }
466 }
```

(End of definition for `\_hook_cmd_if_scannable:NnTF`.)

`\_hook_guess_arg_count>NN`  
`\_hook_guess_arg_count:wN`  
`\_hook_guess_arg_count:nw`

Looks at the parameter text of a macro, and counts the parameters by looking at the number after a `#`, and checking if they are sequential. This macro assumes that all parameters are marked with hashes, and not other characters, and that there is no “trick parameter”.

```

467 <|latexrelease> \IncludeInRelease{2023/06/01}{_hook_guess_arg_count:NN}
468 <|latexrelease> \IncludeInRelease{2023/06/01}{cmd~hooks~with~args}
469 \cs_new_protected:Npn _hook_guess_arg_count:NN #1
470 {
471 \exp_after:wN _hook_guess_arg_count:wN
472 \token_to_meaning:N #1 \s__hook_mark
473 }
474 \exp_last_unbraced:NNNNo
475 \cs_new_protected:Npx _hook_guess_arg_count:wN
476 #1 { \tl_to_str:n { macro: } } #2 \s__hook_mark #3
477 }
```

```

478 \int_set:Nn #3
479 {
480 \exp_not:N __hook_guess_arg_count:nw { 0 } #2
481 \c_hash_str 0 \s__hook_mark
482 }
483 }
484 \use:e
485 { \cs_new:Npn \exp_not:N __hook_guess_arg_count:nw #1 #2 \c_hash_str #3 }
486 {
487 \int_compare:nNnTF { #1 + 1 } = {#3}
488 { __hook_guess_arg_count:nw {#3} }
489 { #1 __hook_use_none_delimit_by_s_mark:w }
490 }
491 \end{macro}
492 \end{macro}
493 \end{macro}
494 \end{macro}
495 \end{macro}

(End of definition for __hook_guess_arg_count:NN, __hook_guess_arg_count:wN, and
__hook_guess_arg_count:nw.)

```

\\_\_hook\_patch\_retokenize:Nnnn Then, if \\_\_hook\_cmd\_if\_scannable:NnTF returned true, we can go on and patch the command.

```

496 \end{macro}
497 \end{macro}
498 \cs_new_protected:Npn __hook_patch_retokenize:Nnnn #1 #2 #3 #4
499 {

```

Here, when patching by retokenization, we can only guess the number of arguments of the macro.

```
500 __hook_guess_arg_count:NN #1 \l__hook_patch_num_args_int
```

Then we redefine the hook to have the right number of arguments. Disabling the hook, undefining the parameter token list then calling \\_\_hook\_make\_usable:nn are enough to redefine the hook to the extent we want. Code stored in the hook and other metadata about it are not lost in the process.

```

501 __hook_disable:n {#2}
502 \cs_undefine:c { c__hook_##2_parameter_tl }
503 __hook_make_usable:nn {#2} { \l__hook_patch_num_args_int }
504 \tl_set:Ne \l__hook_tmpa_tl
505 { \exp_args:Ne \tl_to_str:n { __hook_braced_parameter:n {#2} } }
506 \use:x
507 {
508 \str_replace_all:Nnn \exp_not:N \l__hook_tmpa_tl
509 { ##### } { \c_hash_str }
510 }

```

Then, make make some things \relax to avoid lots of \noexpand below.

```

511 \cs_set_eq:NN \kerneltmpDoNotUse \scan_stop:
512 \cs_set_eq:NN __hook_tmp:w \scan_stop:
513 \use:x
514 {

```

Now we'll define \\_\_hook\_tmp:w such that it splits the \meaning of the macro (#1) into its three parts:

```

#####1. <prefixes>
#####2. <parameter text>
#####3. <replacement text>

```

and arrange that a complete definition, then place the `before` or `after` hooks around the `<replacement text>`: accordingly.

```

515 \cs_set:Npn __hook_tmp:w
516 #####1 \tl_to_str:n { macro: } #####2 -> #####3 \s__hook_mark
517 {
518 #####1 \def \kerneltmpDoNotUse #####2
519 {
520 \str_if_eq:nnT {#3} { before }
521 {
522 \token_to_str:N \UseHookWithArguments {#2}
523 { \int_use:N \l__hook_patch_num_args_int }
524 \l__hook_tmpa_tl
525 }
526 #####3
527 \str_if_eq:nnT {#3} { after }
528 {
529 \token_to_str:N \UseHookWithArguments {#2}
530 { \int_use:N \l__hook_patch_num_args_int }
531 \l__hook_tmpa_tl
532 }
533 }
534 }

```

Now we just have to get the `\meaning` of the command being patched and pass it through the meat grinder above.

```

535 \tl_set:Nx \exp_not:N \l__hook_tmpa_tl
536 { \exp_not:N __hook_tmp:w \token_to_meaning:N #1 \s__hook_mark }
537 }

```

Now rescan with the given catcode settings (overridden by the `\__hook_patch_required_catcodes:`), and implicitly (by using the rescanned token list) carry out the definition from above.

```

538 \tl_rescan:nV { #4 __hook_patch_required_catcodes: } \l__hook_tmpa_tl

```

And to close, copy the newly-defined command into the old name and the patching is finally completed:

```

539 \cs_gset_eq:NN #1 \kerneltmpDoNotUse

```

Finally, update the hook code.

```

540 __hook_update_hook_code:n {#2}
541 }
542 \EndIncludeInRelease
543 \IncludeInRelease{2021/06/01}{__hook_patch_retokenize:Nnnn}
544 \cmd-hooks-with-args
545 \cs_gset_protected:Npn __hook_patch_retokenize:Nnnn #1 #2 #3 #4
546 {
547 \EndIncludeInRelease
548 \cs_set_eq:NN \kerneltmpDoNotUse \scan_stop:
549 \cs_set_eq:NN __hook_tmp:w \scan_stop:

```

```

550 <|latexrelease> {
551 <|latexrelease> \cs_set:Npn __hook_tmp:w
552 <|latexrelease> #####1 \tl_to_str:n { macro: } #####2 -> #####3 \s_hook_mark
553 <|latexrelease> {
554 <|latexrelease> #####1 \def \kerneltmpDoNotUse #####
555 <|latexrelease> {
556 <|latexrelease> \str_if_eq:nnT {#3} { before }
557 <|latexrelease> { \token_to_str:N \UseHook { cmd / #2 / #3 } }
558 <|latexrelease> #####
559 <|latexrelease> \str_if_eq:nnT {#3} { after }
560 <|latexrelease> { \token_to_str:N \UseHook { cmd / #2 / #3 } }
561 <|latexrelease> }
562 <|latexrelease> }
563 <|latexrelease> \tl_set:Nx \exp_not:N \l__hook_tmpa_tl
564 <|latexrelease> { \exp_not:N __hook_tmp:w \token_to_meaning:N #1 \s_hook_mark }
565 <|latexrelease> }
566 <|latexrelease> \tl_rescan:nV {#4 __hook_patch_required_catcodes: } \l__hook_tmpa_tl
567 <|latexrelease> \cs_gset_eq:NN #1 \kerneltmpDoNotUse
568 <|latexrelease> }
569 <|latexrelease> \EndIncludeInRelease

```

(End of definition for `\__hook_patch_retokenize:Nnnn`.)

## 4.6 Messages

```

570 <|latexrelease> \IncludeInRelease{2023/06/01}{wrong-cmd-hook}%
571 <|latexrelease> {Standardise-generic-hook-names}
572 <|latexrelease> \EndIncludeInRelease
573 <|latexrelease> \IncludeInRelease{2021/06/01}{wrong-cmd-hook}%
574 <|latexrelease> {Standardise-generic-hook-names}
575 <|latexrelease> \msg_new:nnnn { hooks } { wrong-cmd-hook }
576 <|latexrelease> {
577 <|latexrelease> Generic~hook~`cmd/#1/#2'~is~invalid.
578 <|latexrelease>% The~hook~should~be~`cmd/#1/before'~or~`cmd/#1/after'.
579 <|latexrelease> }
580 <|latexrelease> {
581 <|latexrelease> You~tried~to~add~a~generic~hook~to~command~\iow_char:N \\#1,~but~`#2'~
582 <|latexrelease> is~an~invalid~component.~Only~`before'~or~`after'~are~allowed.
583 <|latexrelease> }
584 <|latexrelease> \EndIncludeInRelease
585 \msg_new:nnnn { hooks } { cant-patch }
586 {
587 Generic~hooks~cannot~be~added~to~`#1'.
588 }
589 {
590 You~tried~to~add~a~hook~to~`#1',~but~LaTeX~was~unable~to~
591 patch~the~command~because~it~__hook_unpatchable_cases:n {#2}.
592 }
593 \cs_new:Npn __hook_unpatchable_cases:n #1
594 {
595 \str_case:nn {#1}
596 {
597 { undef } { doesn't~exist }
598 { macro } { is~not~a~macro }

```

```

599 { expl3 } { is-a-private-expl3-macro }
600 { retok } { can't-be-retokenized-cleanly }
601 }
602 }

603 <texrelease> \IncludeInRelease{0000/00/00}{ltcmdhooks}%
604 <texrelease> {The~hook~management~system~for~commands}
605 <texrelease>

```

The command `\__hook_cmd_begindocument_code:` is used in an internal hook, so we need to make sure it has a harmless definition after rollback as that will not remove it from the kernel hook.

```

606 <texrelease> \cs_set_eq:NN __hook_cmd_begindocument_code: \prg_do_nothing:
607 <texrelease>
608 <texrelease> \EndModuleRelease
609 \ExplSyntaxOff
610 </2ekernel | texrelease>
611 <@>

```

# File 10

## ltsockets.dtx

### Abstract

This code implements sockets which are places in the code into which predeclared chunks of code (plugs) can be placed. Both the sockets and the plugs are “named” and each socket is assigned exactly one plug at any given time.

## 1 Introduction

A L<sup>A</sup>T<sub>E</sub>X source file is transformed into a typeset document by executing code for each command or environment in the document source. Through various steps this code transforms the input and eventually generates typeset output appearing in a “galley” from which individual pages are cut off in an asynchronous way. This page generating process is normally not directly associated with commands in the input<sup>21</sup> but is triggered whenever the galley has received enough material to form another page (giving current settings).

As part of this transformation input data may get stored in some form and later reused, for example, as part of the output routine processing.

## 2 Configuration of the transformation process

There are three different major methods offered by L<sup>A</sup>T<sub>E</sub>X to configure the transformation process:

- through the template mechanism,
- through the hook mechanism, or
- through sockets and plugs.

They offer different possibilities (with different features and limitations) and are intended for specific use cases, though it is possible to combine them.

### 2.1 The template mechanism

The template mechanism is intended for more complex document-level elements (e.g., headings such as `\section` or environments like `itemize`). The template code implements the overall processing logic for such an element and offers a set of parameters to influence the final result.

The document element is then implemented by (a) selecting a suitable template (there may be more than one available for the kind of document element) and (b) by setting its parameters to desired values. This then forms a so-called instance which is executed when the document element is found in the source.

By altering the parameter values (in a document class or in the document preamble) or, if more drastic layout changes are desired, by selecting a different template and

---

<sup>21</sup>Expects for directives such as `\newpage`.

then adjusting its parameters, a wide variety of layouts can be realized through simple configuration setups without the need to develop new code.

The target audience of this method are therefore document class developers or users who wish to alter an existing layout (implemented by a document class) in certain (minor) ways.

The template mechanism is currently documented as part of the `xtemplate` package and one more elaborate implementation can be found as part of the `latex-lab` code for lists (to be documented further).

## 2.2 The hook mechanism

Hooks are places in the kernel code (or in packages) that offer packages the possibility to inject additional code at specific points in the processing in a controlled way without the need to replace the existing code block (and thereby overwriting modifications/extensions made by other packages). The target audience is therefore mainly package developers, even though some hooks can be useful for document authors.

Obviously, what can reasonably be added into a hook depends on the individual hook (hopefully documented as part of the hook documentation), but in general the idea behind hooks is that more than one package could add code into the hook at the same time. Perhaps the most famous hook (that L<sup>A</sup>T<sub>E</sub>X had for a very long time) is `begindocument` into which many packages add code to through `\AtBeginDocument{\{code\}}` (which is nowadays implemented as a shorthand for `\AddToHook{begindocument}{\{code\}}`). To resolve possible conflicts between injections by different packages there is a rule mechanism by which code chunks in a hook can be ordered in a certain way and by which incompatible packages can be detected if a resolution is impossible.

In contrast to template code, there is no standard configuration method through parameters for hooks, i.e., the code added to a hook “is” the configuration. If it wants to provide for configuration through parameters it has to also provide its own method to set such parameters in some way. However, in that case it is likely that using a hook is not the right approach and the developer better calls a template instance instead which then offers configuration through a key/value interface.

In most cases, hooks do not take any arguments as input. Instead, the data that they can (and are allowed to) access depends on the surrounding context.

For example, the various hooks available during the page shipout process in L<sup>A</sup>T<sub>E</sub>X’s output routine can (and have to) access the accumulated page material stored in a box named `\ShipoutBox`. This way, code added to, say, the `shipout/before` hook could access the page content, alter it, and then write it back into `\ShipoutBox` and any other code added to this hook could then operate on the modified content. Of course, for such a scheme to work the code prior to executing the hook would need to setup up data in appropriate places and the hook documentation would need to document what kind of storage can be accessed (and possibly altered) by the hook.

There are also hooks that take arguments (typically portions of document data) and in that case the hook code can access these arguments through #1, #2, etc.

The hook mechanism is documented in `lthooks-doc.pdf`.

## 2.3 The socket mechanism

In some cases there is code that implements a certain programming logic (for example, combining footnotes, floats, and the text for the current page to be shipped out) and

if this logic should change (e.g., footnotes to be placed above bottom floats instead of below) then this whole code block needs to be replaced with different code.

In theory, this could be implemented with templates, i.e., the code simply calls some instance that implements the logic and that instance is altered by selecting a different templates and/or adjusting their parameters. However, in many cases customization through parameters is overkill in such a case (or otherwise awkward, because parameterization is better done on a higher level instead of individually for small blocks of code) and using the template mechanism just to replace one block of code with a different one results in a fairly high performance hit. It is therefore usually not a good choice.

In theory, it would also be possible to use a hook, but again that is basically a misuse of the concept, because in this use case there should never be more than one block of code inside the hook; thus, to alter the processing logic one would need to set up rules that replace code rather than (as intended) execute all code added to the hook.

For this reason L<sup>A</sup>T<sub>E</sub>X now offers a third mechanism: “sockets” into which one can place exactly one code block — a “plug”.

In a nutshell: instead of having a fixed code block somewhere as part of the code, implementing a certain programming logic there is a reference to a named socket at this point. This is done by first declaring the named socket with:

```
\NewSocket{\langle socket-name\rangle}{\langle number-of-inputs\rangle}
```

This is then referenced at the point where the replaceable code block should be executed with:

```
\UseSocket{\langle socket-name\rangle}
```

or, if the socket should take a number of inputs (additional arguments beside the name) with

```
\UseSocket{\langle socket-name\rangle}{\langle arg_1\rangle}...{\langle arg_{\langle number-of-inputs\rangle}\rangle}
```

In addition, several code blocks (a.k.a. plugs) implementing different logic for this socket are set up, each with a declaration of the form:

```
\NewSocketPlug{\langle socket-name\rangle}{\langle socket-plug-name\rangle}{\langle code\rangle}
```

Finally, one of them is assigned to the socket:

```
\AssignSocketPlug{\langle socket-name\rangle}{\langle socket-plug-name\rangle}
```

If the programming logic should change, then all that is necessary is to make a new assignment with `\AssignSocketPlug` to a different `\langle socket-plug-name\rangle`. This assignment obeys scope so that an environment can alter a socket without the need to restore the previous setting manually.

If the socket takes inputs, then those need to be provided to `\UseSocket` and in that case they can be referenced in the `\langle code\rangle` argument of `\NewSocketPlug` with #1, #2, etc.

In most cases a named socket is used only in a single place, but there is, of course, nothing wrong with using it in several places, as long as the code in all places is supposed to change in the same way.

### 2.3.1 Examples

We start by declaring a new socket named `foo` that expects two inputs:

```
\NewSocket{foo}{2}
```

Such a declaration has to be unique across the whole L<sup>A</sup>T<sub>E</sub>X run. Thus, if another package attempts to use the same name (regardless of the number of inputs) it will generate an error:

```
\NewSocket{foo}{2}
\NewSocket{foo}{1}
```

Both declarations would therefore produce:

```
! LaTeX socket Error: Socket 'foo' already declared!
```

You also get an error if you attempt to declare some socket plug and the socket name is not yet declared, e.g.,

```
\NewSocketPlug{baz}{undeclared}{some code}
```

generates

```
! LaTeX socket Error: Socket 'baz' undeclared!
```

Setting up plugs for the socket is done like this:

```
\NewSocketPlug{foo}{plug-A}
 {\begin{quote}\itshape foo-A: #1!#2\end{quote}}
\NewSocketPlug{foo}{plug-B}
 {\begin{quote}\sffamily foo-B: #22\end{quote}}
```

This will set up the plugs `plug-A` and `plug-B` for this socket.

We still have to assign one or the other to the socket, thus without doing that the line

```
\UseSocket{foo}{hello}{world}
```

produces nothing because the default plug for sockets with 2 inputs is `noop` (which grabs the additional arguments and throws them away).<sup>22</sup>

So let's do the assignment

```
\AssignSocketPlug{foo}{plug-A}
```

and then

```
\UseSocket{foo}{hello}{world}
```

will properly typeset

*foo-A: hello!world*

and after

```
\AssignSocketPlug{foo}{plug-B}
```

---

<sup>22</sup>If socket `foo` would have been a socket with one input, then the default plug would be `identity`, in which case the socket input would remain without braces and gets typeset!

and another call to

```
\UseSocket{foo}{hello}{world}
```

we get

```
foo-B: world2
```

If we attempt to assign a plug that was not defined, e.g.,

```
\AssignSocketPlug{foo}{plug-C}
```

then we get an error during the assignment

```
! LaTeX socket Error: Plug 'plug-C' for socket 'foo' undeclared!
```

and the previous assignment remains in place.

To see what is known about a socket and its plugs you can use `\ShowSocket` or `\LogSocket` which displays information similar to this on the terminal or in the transcript file:

```
Socket foo:
 number of inputs = 2
 available plugs = noop, plug-A, plug-B
 current plug = plug-B
 definition = \long macro:#1#2->\begin {quote}\sffamily
 foo-B: #2\textsuperscript {2}\end {quote}
```

### 2.3.2 Details and semantics

In this section we collect some normative statements.

- From a functional point of view sockets are like simple T<sub>E</sub>X macros, i.e., they expect 0 to 9 mandatory arguments (the socket inputs) and get replaced by their “expansion”
- A socket is “named” and the name consists of ASCII letters [a-z], [A-Z], [0-9], [-/\@] only
- Socket names have to be unique, i.e., there can be only one socket named `\name`. This is ensured by declaring each socket with `\NewSocket`.

However, there is no requirement that sockets and hook names have to be different. In fact, if a certain action that could otherwise be specified as hook code has to be executed always last (or first) one could ensure this by placing a socket (single action) after a hook (or vice versa) and using the same name to indicate the relationship, e.g.,

```
\UseHook{foo} % different package can add code here
\UseSocket{foo} % only one package can assign a plug
```

This avoids the need to order the hook code to ensure that something is always last.

- Best practice naming conventions are ... *to be documented*

- A socket has documented inputs which are
  - the positional arguments (if any) with a description of what they contain when used
  - implicit data (registers and other 2e/expl3 data stores) that the socket is allowed to make use of, with a documented description of what they contain (if relevant for the task at hand—no need to describe the whole L<sup>A</sup>T<sub>E</sub>X universe)
  - information about the state of the T<sub>E</sub>X engine (again when relevant), e.g. is called in mmode or vmode or in the output routine or ...
  - ... anything missing?
- A socket has documented results/outputs which can be
  - what kind of data it should write to the current list (if that is part of its task)
  - what kind of registers and other 2e/expl3 data stores it should modify and in what way
  - what kind of state changes it should do (if any)
  - ... *anything else?*
- At any time a socket has one block of code (a plug :-)) associated with it. Such code is itself named and the association is done by linking the socket name to the code name (putting a plug into the socket).
- The name of a plug consists of ASCII letters [a-z], [A-Z], [0-9], [-@] only.
- Socket plug names have to be unique within on a per socket basis, but it is perfectly allowed (and sensible in some cases) to use the same plug name with different sockets (where based on the sockets' purposes, different actions may be associated with the plug name). For example `noop` is a plug name declared for every socket, yet its action “grab the socket inputs and throw them away” obviously differs depending on how many inputs the socket has.
- When declaring a plug it is stated for which socket it is meant (i.e., its code can only be used with that socket). This means that the same plug name can be used with different sockets referring to different code in each case.
- Configuration of a socket can only be done by linking different code to it. Nevertheless the code linked to it can provide its own means of configuration (but this is outside of the spec).
- Technically execution of a socket (`\UseSocket`) involves
  - doing any house keeping (like writing debugging info, ...);
  - looking up the current code association (what plug is in the socket);
  - executing this code which will pick up the mandatory arguments (happens at this point, not before), i.e., it is like calling a csname defined with  
`\def\foo#1#2...{...#1...#2...}`
  - do some further house keeping (if needed).
- A socket is typically only used in one place in code, but this is not a requirement, i.e., if the same operation with the same inputs need to be carried out in several places the same named socket can be used.

### 2.3.3 Command syntax

We give both the L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  and the L3 programming layer command names.

---

```
\NewSocket \NewSocket {\langle socket-name\rangle} {\langle number-of-inputs\rangle}
\socket_new:nn \socket_new:nn {\langle socket-name\rangle} {\langle number-of-inputs\rangle}
```

---

Declares a new socket with name *<socket-name>* having *<number-of-inputs>* inputs. There is automatically a plug *noop* declared for it, which does nothing, i.e., it gobbles the socket inputs (if any). This is made the default plug except for sockets with one input which additionally define the plug *identity* and assign that as their default.

This *identity* plug simply returns the socket input without its outer braces. The use case for this plug are situations like this:

```
\UseSocket{tagsupport/footnote}{\langle code\rangle}
```

If tagging is not active and the socket contains the plug *identity* then this returns *\langle code\rangle* without the outer braces and to activate tagging all that is necessary is to change the plug to say *tagpdf* so that it surrounds *\langle code\rangle* by some tagging magic. This is the most common use case for sockets with one input, which is why they have this special default.

The socket documentation should describe its purpose, its inputs and the expected results as discussed above.

The declaration is only allowed at top-level, i.e., not inside a group.

---

```
\NewSocketPlug \NewSocketPlug {\langle socket-name\rangle} {\langle socket-plug-name\rangle} {\langle code\rangle}
\socket_new_plug:nnn \socket_new_plug:nnn {\langle socket-name\rangle} {\langle socket-plug-name\rangle} {\langle code\rangle}
\socket_set_plug:nnn \socket_set_plug:nnn {\langle socket-name\rangle} {\langle socket-plug-name\rangle} {\langle code\rangle}
```

---

Declares a new plug for socket *<socket-name>* that runs *<code>* when executing. It complains if the plug was already declared previously.

The form *\socket\_set\_plug:nnn* changes an existing plug. As this should normally not be necessary, we currently have only an L3 layer name for the few cases it might be useful.

The declarations can be made inside a group and obey scope, i.e., they vanish if the group ends.

---

```
\AssignSocketPlug \AssignSocketPlug {\langle socket-name\rangle} {\langle socket-plug-name\rangle}
\socket_assign_plug:nn \socket_assign_plug:nn {\langle socket-name\rangle} {\langle socket-plug-name\rangle}
```

---

Assigns the plug *<socket-plug-name>* to the socket *<socket-name>*. It errors if either socket or plug is not defined.

The assignment is local, i.e., it obeys scope.

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\UseSocket</code>       | <code>\UseSocket {&lt;socket-name&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>\socket_use:nw</code>   | <code>\socket_use:nnn {&lt;socket-name&gt;} {&lt;socket-arg<sub>1</sub>&gt;} {&lt;socket-arg<sub>2</sub>&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>\socket_use:n</code>    | Executes the socket <code>&lt;socket-name&gt;</code> by retrieving the <code>&lt;code&gt;</code> of the current plug assigned to the socket. This is the only command that would appear inside macro code in packages.                                                                                                                                                                                                                                                                                                                                                                              |
| <code>\socket_use:nn</code>   | For performance reasons there is no explicit check that the socket was declared!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\socket_use:nnn</code>  | The different L3 programming layer commands are really doing the same thing: they grab as many arguments as defined as inputs for the socket and then pass them to the plug. The different names are only there to make the code more readable, i.e., to indicate how many arguments are grabbed in total (note that no runtime check is made to verify that this is actually true). We only provide them for sockets with up to 3 inputs (most likely those with zero or one input would have been sufficient). If you happen to have a socket with more inputs, use <code>\socket_use:nw</code> . |
| <code>\socket_use:nnnn</code> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                                        |                                                               |
|----------------------------------------|---------------------------------------------------------------|
| <code>\socket_use_expandable:nw</code> | <code>* \socket_use_expandable:n {&lt;socket-name&gt;}</code> |
| <code>\socket_use_expandable:n</code>  | <code>*</code>                                                |

Fully expandable variant of `\socket_use:n`. This can be used in macro code to retrieve code from sockets which need to appear in an expandable context.

This usually requires the plug to only contain expandable code and should therefore only be used for sockets which are clearly documented to be used in an expandable context. This command does not print any debugging info when `\DebugSocketsOn` is active and should therefore be avoided whenever possible.

For performance reasons there is no explicit check that the socket was declared!

|                             |                                                                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\ShowSocket</code>    | <code>\ShowSocket {&lt;socket-name&gt;}</code>                                                                                                                            |
| <code>\LogSocket</code>     | <code>\socket_show:n {&lt;socket-name&gt;}</code>                                                                                                                         |
| <code>\socket_show:n</code> | Displays information about the socket <code>&lt;socket-name&gt;</code> and its state then stops and waits for further instructions — at the moment some what rudimentary. |
| <code>\socket_log:n</code>  | <code>\LogSocket</code> and <code>\socket_log:n</code> only differ in that they don't stop.                                                                               |

|                                 |                                                   |
|---------------------------------|---------------------------------------------------|
| <code>\DebugSocketsOn</code>    | <code>\DebugSocketsOn ... \DebugSocketsOff</code> |
| <code>\DebugSocketsOff</code>   |                                                   |
| <code>\socket_debug_on:</code>  | Turns debugging of sockets on or off.             |
| <code>\socket_debug_off:</code> |                                                   |

### 2.3.4 Rationale for error handling

The errors during the declarations are produced to help with typos—after all, such declarations might be part of a document preamble (not that likely, but possible). However, `\UseSocket` is not doing much checking, e.g.,

```
\UseSocket{misplaced-socket}{hello}{world}
```

will generate a rather low-level error and then typesets “helloworld” because there is no dedicated runtime check if `misplaced-socket` is a known socket.

The reason is that if the misspelling is in the code, then this is a programming error in the package and for speed reasons L<sup>A</sup>T<sub>E</sub>X does not repeatedly make runtime checks for coding errors unless they can or are likely to be user introduced.

### 3 The Implementation

The implementation of the socket mechanism should be (partially) redone and we should probably store the different code chunks in a property list so that we can have a decent `\ShowSocket` command that shows the available alternatives.

TODO: *implement?*

```
1 <*2ekernel | latexrelease>
2 \ExplSyntaxOn
3 <@@=socket>
4 <latexrelease> \NewModuleRelease{2023/11/01}{ltsockets}
5 <latexrelease> {The-socket-management-system}
```

#### 3.1 Debugging the socket structures

Code and commands in this section are not final, it needs more experimentation to see what kind of tracing information is going to be useful in practice. For now the tracing is mainly meant to be used for code testing and not so much for application testing.

It is quite likely that the commands and the behavior of the tracing might change in the future once we gained some experience with it.

`\g__socket_debug_bool` Holds the current debugging state.

```
6 \bool_new:N \g__socket_debug_bool
```

(End of definition for `\g__socket_debug_bool`.)

```
\socket_debug_on: Turns debugging on and off by redefining __socket_debug:n and __socket_debug_-term:n. By default they do nothing.
\socket_debug_off:
__socket_debug:n
__socket_debug_term:n
__socket_debug_gset:
7 \cs_new_eq:NN __socket_debug:n \use_none:n
8 \cs_new_eq:NN __socket_debug_term:n \use_none:n
9 \cs_new_protected:Npn \socket_debug_on:
10 {
11 \bool_gset_true:N \g__socket_debug_bool
12 __socket_debug_gset:
13 }
14 \cs_new_protected:Npn \socket_debug_off:
15 {
16 \bool_gset_false:N \g__socket_debug_bool
17 __socket_debug_gset:
18 }
19 \cs_new_protected:Npn __socket_debug_gset:
20 {
21 \cs_gset_protected:Npx __socket_debug:n ##1
22 { \bool_if:NT \g__socket_debug_bool {##1} }
23 \cs_gset_protected:Npx __socket_debug_term:n ##1
24 { \bool_if:NT \g__socket_debug_bool
25 { \iow_term:x { ^~J [Sockets]~ ==>~ ##1} } }
26 }
```

(End of definition for `\socket_debug_on:` and others. These functions are documented on page 335.)

### 3.2 The L3 layer commands

`\socket_new:nn` Declaring a socket creates a str to hold the name (a pointer) to the code that should be used when the socket is executed, and an integer to hold the number of inputs of that socket. Initially, an “empty” code chunk is created and assigned so the socket does nothing by default other than swallowing its inputs (if any).

```

27 \cs_new_protected:Npn \socket_new:nn #1 #2 {
28 \str_if_exist:cTF { l__socket_#1_plug_str }
29 {
30 \msg_error:nnn { socket } { already-declared } {#1}
31 }
32 }
```

We only support declarations on top-level.

```

33 \int_if_zero:nTF \tex_currentgrouplevel:D
34 {
35 \str_new:c { l__socket_#1_plug_str }
36 \seq_new:c { l__socket_#1_plugs_seq }
37 \int_const:cn { c__socket_#1_args_int } {#2}
38 \socket_new_plug:nnn {#1} { noop } {}
39 \int_compare:nNnTF {#2} = 1
40 {
41 \socket_new_plug:nnn {#1} { identity } {##1}
42 \socket_assign_plug:nn {#1} { identity }
43 }
44 { \socket_assign_plug:nn {#1} { noop } }
45 _socket_debug_term:n
46 { Socket~ '#1'~ declared~ with~ #2~ input(s) }
47 }
48 {
49 \msg_error:nn { socket } { not-top-level }
50 }
51 }
52 }
```

(End of definition for `\socket_new:nn`. This function is documented on page 334.)

`\socket_log:n` Show the current state of the socket — for now this is just a quick draft and should be redone and extended.  
`\socket_show:n`

```

53 \cs_new_protected:Npn \socket_log:n #1 {
54 \typeout{ Socket~ #1:}
55 \str_if_exist:cTF { l__socket_#1_plug_str }
56 {
57 \typeout{ \@spaces number~ of~ inputs~ ==
58 \int_use:c { c__socket_#1_args_int } }
59 \typeout{ \@spaces available~plugs~ ==
60 \seq_use:cnnn { l__socket_#1_plugs_seq }{,}{,}{,}{,}{,} }
61 \typeout{ \@spaces current~ plug~ ==
62 \str_use:c { l__socket_#1_plug_str } }
63 \typeout{ \@spaces definition~ ==
64 \cs_meaning:c
65 { __socket_#1_plug_ \str_use:c { l__socket_#1_plug_str } :w } }
66 \typeout{ }
67 }
68 }
```

If we are showing a socket it is not an error if it doesn't exist.

```
69 \typeout { Socket~ is~ not~ declared! }
70 }
71 }
```

And here the version that stops:

```
72 \cs_new_protected:Npn \socket_show:n #1 {\socket_log:n {#1} \errmessage{}}
```

(End of definition for `\socket_log:n` and `\socket_show:n`. These functions are documented on page 335.)

`\socket_new_plug:nnn` Declaring a code for a socket is just making a definition, taking the number of arguments from the saved int.

```
73 \cs_new_protected:Npn \socket_new_plug:nnn #1#2#3 {
74 \str_if_exist:cTF { l__socket_#1_plug_str }
75 {
76 \cs_if_exist:cTF { __socket_#1_plug_#2:w }
77 {
78 \msg_error:nnnn { socket } { plug-already-declared } {#1} {#2}
79 }
80 }
81 \cs_generate_from_arg_count:cNnn
82 { __socket_#1_plug_#2:w }
83 \cs_new:Npn
84 { \int_use:c { c__socket_#1_args_int } }
85 {#3}
```

This is a new declaration so we add the name to a seq for the debugging info.

```
86 \seq_put_right:cn { l__socket_#1_plugs_seq } {#2}
87 __socket_debug_term:n
88 { Plug~ '#2'~ for~ socket~ '#1'~ declared. }
89 }
90 }
91 {
92 \msg_error:nnn { socket } { undeclared } {#1}
93 }
94 }
```

Changing the plug of an existing socket is rather similar, except that we don't have to deal with adding it to the debugging sequence.

```
95 \cs_new_protected:Npn \socket_set_plug:nnn #1#2#3 {
96 \str_if_exist:cTF { l__socket_#1_plug_str }
97 {
98 \cs_if_exist:cTF { __socket_#1_plug_#2:w }
99 {
100 \cs_generate_from_arg_count:cNnn
101 { __socket_#1_plug_#2:w }
102 \cs_set:Npn
103 { \int_use:c { c__socket_#1_args_int } }
104 {#3}
105 __socket_debug_term:n
106 { Plug~ '#2'~ for~ socket~ '#1'~ changed. }
107 }
108 }
109 \msg_error:nnnn { socket } { plug-undeclared } {#1} {#2}
```

```

110 }
111 }
112 {
113 \msg_error:n { socket } { undeclared } {#1}
114 }
115 }

```

(End of definition for `\socket_new_plug:nnn` and `\socket_set_plug:nnn`. These functions are documented on page 334.)

**\socket\_assign\_plug:nn** Assigning a plug to a socket just changes the name in the socket string. The assignment is local to the current group.

```

116 \cs_new_protected:Npn \socket_assign_plug:nn #1 #2 {
117 \str_if_exist:cTF { l__socket_#1_plug_str } {
118 \cs_if_exist:cTF { __socket_#1_plug_#2:w } {
119 __socket_debug_term:n
120 { Replacing~ plug~ '\str_use:c { l__socket_#1_plug_str }'~
121 with~ '#2'~ in~ socket~ '#1'. }
122 \str_set:cn { l__socket_#1_plug_str } {#2}
123 }
124 }
125 \msg_error:nnnn { socket } { plug-undeclared } {#1} {#2}
126 }
127 }
128 }
129 }
130 { \msg_error:nn { socket } { undeclared } {#1} }
131 }

```

(End of definition for `\socket_assign_plug:nn`. This function is documented on page 334.)

**\socket\_use:nw**  
**\socket\_use:n**  
**\socket\_use:nn**  
**\socket\_use:nnn**  
**\socket\_use:nnnn**

And using it is more or less a `\use:c` so very lightweight. We do not add a runtime check for speed reasons!

This command is named `\socket_use:nw` because we don't know how many inputs the socket has until we have looked at the socket name (in argument `#1`). But, of course, the developer knows so we also offer a few aliases `\socket_use:nn`, etc. so that one can indicate the correct number of arguments (socket inputs plus one) in the L3 layer code.

```

132 \cs_new_protected:Npn \socket_use:nw #1 {
133 __socket_debug_term:n
134 { Socket~ '#1'~ containing~ plug~
135 '\str_use:c { l__socket_#1_plug_str }'~ used. }
136 \use:c { __socket_#1_plug_ \str_use:c { l__socket_#1_plug_str } :w }
137 }

```

To make code a bit more readable we also define functions that indicate how many arguments are picked up. However, this is just for code documentation: internally they all do the same and the number of arguments isn't checked by default.

```

138 \cs_new_eq:NN \socket_use:n \socket_use:nw % socket with no inputs
139 \cs_new_eq:NN \socket_use:nn \socket_use:nw % socket with one input
140 \cs_new_eq:NN \socket_use:nnn \socket_use:nw % socket with two inputs
141 \cs_new_eq:NN \socket_use:nnnn \socket_use:nw % socket with three inputs

```

The above commands could be changed to check how many inputs the socket is declared with (for example, when checking is in force).

**TODO:** *Implement?*

(End of definition for \socket\_use:nw and others. These functions are documented on page 335.)

\socket\_use\_expandable:nw The same as the non-expandable code, except for the missing debug output.

```
142 \cs_new:Npn \socket_use_expandable:nw #1 {
143 \use:c { __socket_#1_plug_ \str_use:c { l__socket_#1_plug_str } :w }
144 }
145 \cs_new_eq:NN \socket_use_expandable:n \socket_use_expandable:nw % socket with no inputs
```

(End of definition for \socket\_use\_expandable:nw and \socket\_use\_expandable:n. These functions are documented on page 335.)

### 3.3 Error messages

```
146 \msg_new:nnnn { socket } { already-declared }
147 { Socket~ '#1'~ already~ declared! }
148 { A~ socket~ can~ only~ be~ declared~ once.~ The~ name~ '#1'~ is~
149 already~ taken.~ Use~ \ShowSocket{#1}~ to~ see~ its~ definition. }
150
151 \msg_new:nnnn { socket } { undeclared }
152 { Socket~ '#1'~ undeclared! }
153 { You~ tried~ to~ use~ a~ socket~ that~ was~ not~ declared~ before. }
154
155 \msg_new:nnnn { socket } { not-top-level }
156 { Sockets~ can~ only~ be~ declared~ at~ top-level! }
157 { It~ is~ not~ allowed~ to~ declare~ sockets~ inside~ a~
158 group.~ Move~ the~ declaration~ to~ the~ top-level. }
159 \msg_new:nnnn { socket } { plug-already-declared }
160 { Plug~ '#2'~ for~ socket~ '#1'~ already~ declared! }
161 { You~ can't~ change~ an~ existing~ plug~ with~ \NewSocketPlug~ and~ it~
162 is~ normally~ not~ sensible~ to~ do~ so.~ Use~ the~ L3~ programming~
163 layer~ function~ \socket_set_plug:nnn~ if~ you~ really~ have~ to. }
164
165 \msg_new:nnnn { socket } { plug-undeclared }
166 { Plug~ '#2'~ for~ socket~ '#1'~ undeclared! }
167 { The~ plug~ name~ is~ unknown.~ Is~ the~ name~ misspelled~ or~ did~ you~
168 intend~ to~ assign~ it~ to~ a~ different~ socket? }
169 \prop_gput:Nnn \g_msg_module_type_prop { socket } { LaTeX }
```

### 3.4 The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> interface commands

\NewSocket As we expect that there are existing L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> packages that may want to make use of the socket mechanism, we provide 2e names for most of the commands.

```
170 \cs_new_eq:NN \NewSocket \socket_new:nn
171 \cs_new_eq:NN \ShowSocket \socket_show:n
172 \cs_new_eq:NN \LogSocket \socket_log:n
173 \cs_new_eq:NN \NewSocketPlug \socket_new_plug:nnn
174 \cs_new_eq:NN \AssignSocketPlug \socket_assign_plug:nn
175 \cs_new_eq:NN \UseSocket \socket_use:nw
176 \cs_new_eq:NN \DebugSocketsOn \socket_debug_on:
177 \cs_new_eq:NN \DebugSocketsOff \socket_debug_off:
```

(End of definition for \NewSocket and others. These functions are documented on page 334.)

```
178 %
179 <|latexrelease>\IncludeInRelease{0000/00/00}{ltsockets}
180 <|latexrelease> {The~socket~management~(undo)}%
181 <|latexrelease>
182 <|latexrelease>\let \NewSocket \@undefined
183 <|latexrelease>\let \ShowSocket \@undefined
184 <|latexrelease>\let \LogSocket \@undefined
185 <|latexrelease>
186 <|latexrelease>\let \NewSocketPlug \@undefined
187 <|latexrelease>\let \AssignSocketPlug \@undefined
188 <|latexrelease>\let \UseSocket \@undefined
189 <|latexrelease>
190 <|latexrelease>\let \DebugSocketsOn \@undefined
191 <|latexrelease>\let \DebugSocketsOff \@undefined
192 <|latexrelease>
193 <|latexrelease>\EndModuleRelease
194 \ExplSyntaxOff
195 </2ekernel | latexrelease>
```

Reset module prefix:

```
196 <@@=>
```

# File 11

## lttemplates.dtx

### 1 Introduction

There are three broad “layers” between putting down ideas into a source file and ending up with a typeset document. These layers of document writing are

1. authoring of the text with mark-up;
2. document layout design;
3. implementation (with `TeX` programming) of the design.

We write the text as an author, and we see the visual output of the design after the document is generated; the `TeX` implementation in the middle is the glue between the two.

`LATEX`’s greatest success has been to standardise a system of mark-up that balances the trade-off between ease of reading and ease of writing to suit almost all forms of technical writing. It’s other original strength was a good background in typographical design; while the standard `LATEX 2 $\epsilon$`  classes look somewhat dated now in terms of their visual design, their typography is generally sound (barring the occasional minor faults).

However, `LATEX 2 $\epsilon$`  has always lacked a standard approach to customising the visual design of a document. Changing the looks of the standard classes involved either:

- Creating a new version of the implementation code of the class and editing it.
- Loading one of the many packages to customise certain elements of the standard classes.
- Loading a completely different document class, such as `KOMA-Script` or `memoir`, that allows easy customisation.

All three of these approaches have their drawbacks and learning curves.

The idea behind `lttemplates` is to cleanly separate the three layers introduced at the beginning of this section, so that document authors who are not programmers can easily change the design of their documents. `lttemplates` also makes it easier for `LATEX` programmers to provide their own customisations on top of a pre-existing class.

### 2 What is a document?

Besides the textual content of the words themselves, the source file of a document contains mark-up elements that add structure to the document. These elements include sectional divisions, figure/table captions, lists of various sorts, theorems/proofs, and so on. The list will be different for every document that can be written.

Each element can be represented logically without worrying about the formatting, with mark-up such as `\section`, `\caption`, `\begin{enumerate}` and so on. The output of each one of these document elements will be a typeset representation of the information marked up, and the visual arrangement and design of these elements can vary widely in producing a variety of desired outcomes.

For each type of document element, there may be design variations that contain the same sort of information but present it in slightly different ways. For example, the difference between a numbered and an unnumbered section, `\section` and `\section*`, or the difference between an itemised list or an enumerated list.

There are three distinct layers in the definition of “a document” at this level

1. semantic elements such as the ideas of sections and lists;
2. a set of design solutions for representing these elements visually;
3. specific variations for these designs that represent the elements in the document.

In the parlance of the template system, these are called types, templates, and instances, and they are discussed below in sections 4, 5, and 7, respectively.

### 3 Types, templates, and instances

By formally declaring documents to be composed of mark-up elements grouped into types, which are interpreted and typeset with a set of templates, each of which has one or more instances with which to compose each and every semantic unit of the text, we can cleanly separate the components of document construction.

All of the structures provided by the template system are global, and do not respect T<sub>E</sub>X grouping.

### 4 Template types

An *template type* (sometimes just “type”) is an abstract idea of a document element that takes a fixed number of arguments corresponding to the information from the document author that it is representing. A sectioning type, for example, might take three inputs: “title”, “short title”, and “label”.

Any given document class will define which types are to be used in the document, and any template of a given type can be used to generate an instance for the type. (Of course, different templates will produce different typeset representations, but the underlying content will be the same.)

---

```
\NewTemplateType \NewTemplateType {\langle template type \rangle} {\langle no. of args \rangle}
```

This function defines an *⟨template type⟩* taking *⟨number of arguments⟩*, where the *⟨type⟩* is an abstraction as discussed above. For example,

```
\NewTemplateType{sectioning}{3}
```

creates a type “sectioning”, where each use of that type will need three arguments.

### 5 Templates

A *template* is a generalised design solution for representing the information of a specified type. Templates that do the same thing, but in different ways, are grouped together by their type and given separate names. There are two important parts to a template:

- the parameters it takes to vary the design it is producing;

| Key-type                       | Description of input                                        |
|--------------------------------|-------------------------------------------------------------|
| <code>boolean</code>           | <code>true</code> or <code>false</code>                     |
| <code>choice{⟨choices⟩}</code> | A list of pre-defined <code>⟨choices⟩</code>                |
| <code>commalist</code>         | A comma-separated list                                      |
| <code>function{⟨N⟩}</code>     | A function definition with $N$ arguments ( $N$ from 0 to 9) |
| <code>instance{⟨name⟩}</code>  | An instance of type <code>⟨name⟩</code>                     |
| <code>integer</code>           | An integer or integer expression                            |
| <code>length</code>            | A fixed length                                              |
| <code>muskip</code>            | A math length with shrink and stretch components            |
| <code>real</code>              | A real (floating point) value                               |
| <code>skip</code>              | A length with shrink and stretch components                 |
| <code>tokenlist</code>         | A token list: any text or commands                          |

Table 1: Key-types for defining template interfaces with `\DeclareTemplateInterface`.

- the implementation of the design.

As a document author or designer does not care about the implementation but rather only the interface to the template, these two aspects of the template definition are split into two independent declarations, `\DeclareTemplateInterface` and `\DeclareTemplateCode`.

---

`\DeclareTemplateInterface` `\DeclareTemplateInterface`  
 $\{\langle type \rangle\} \{\langle template \rangle\} \{\langle no. of args \rangle\}$   
 $\{\langle key list \rangle\}$

A `⟨template⟩` interface is declared for a particular `⟨type⟩`, where the `⟨number of arguments⟩` must agree with the type declaration. The interface itself is defined by the `⟨key list⟩`, which is itself a key–value list taking a specialized format:

```

⟨key1⟩ ":" ⟨key type1⟩ ","
⟨key2⟩ ":" ⟨key type2⟩ ","
⟨key3⟩ ":" ⟨key type3⟩ "=" ⟨default3⟩ ","
⟨key4⟩ ":" ⟨key type4⟩ "=" ⟨default4⟩ ","
...

```

Each `⟨key⟩` name should consist of ASCII characters, with the exception of `,`, `=` and `⋮`. The recommended form for key names is to use lower case letters, with dashes to separate out different parts. Spaces are ignored in key names, so they can be included or missed out at will. Each `⟨key⟩` must have a `⟨key type⟩`, which defined the type of input that the `⟨key⟩` requires. A full list of key types is given in Table 1. Each key may have a `⟨default⟩` value, which will be used in by the template if the `⟨key⟩` is not set explicitly. The `⟨default⟩` should be of the correct form to be accepted by the `⟨key type⟩` of the `⟨key⟩`: this is not checked by the code. Expressions for numerical values are evaluated when the template is used, thus for example values given in terms of `em` or `ex` will be set respecting the prevailing font.

---

```
\KeyValue \KeyValue {\langle key name\rangle}
```

---

There are occasions where the default (or value) for one key should be taken from another. The `\KeyValue` function can be used to transfer this information without needing to know the internal implementation of the key:

```
\DeclareTemplateInterface { type } { template } { no. of args }
{
 key-name-1 : key-type = value ,
 key-name-2 : key-type = \KeyValue { key-name-1 },
 ...
}
```

---

```
\DeclareTemplateCode \DeclareTemplateCode
 {\langle type\rangle} {\langle template\rangle} {\langle no. of args\rangle}
 {\langle key bindings\rangle} {\langle code\rangle}
```

---

The relationship between a templates keys and the internal implementation is created using the `\DeclareTemplateCode` function. As with `\DeclareTemplateInterface`, the `\langle template\rangle` name is given along with the `\langle type\rangle` and `\langle number of arguments\rangle` required. The `\langle key bindings\rangle` argument is a key–value list which specifies the relationship between each `\langle key\rangle` of the template interface with an underlying `\langle variable\rangle`.

```
\langle key1\rangle " = " \langle variable1\rangle,
\langle key2\rangle " = " \langle variable2\rangle,
\langle key3\rangle " = " global \langle variable3\rangle,
\langle key4\rangle " = " global \langle variable4\rangle,
...
```

With the exception of the choice, code and function key types, the `\langle variable\rangle` here should be the name of an existing L<sup>A</sup>T<sub>E</sub>X3 register. As illustrated, the key word “global” may be included in the listing to indicate that the `\langle variable\rangle` should be assigned globally. A full list of variable bindings is given in Table 2.

The `\langle code\rangle` argument of `\DeclareTemplateCode` is used as the replacement text for the template when it is used, either directly or as an instance. This may therefore accept arguments #1, #2, *etc.* as detailed by the `\langle number of arguments\rangle` taken by the type.

---

```
\AssignTemplateKeys \AssignTemplateKeys
```

---

In the final argument of `\DeclareTemplateCode` the assignment of keys defined by the template may be delayed by including the command `\AssignTemplateKeys`. If this is *not* present, keys are assigned immediately before the template code. If `\AssignTemplateKeys` is present, assignment is delayed until this point. Note that the command must be *directly* present in the code, not placed within a nested command/macro.

---

```
\DeclareTemplateCopy \DeclareTemplateCopy
 {\langle type\rangle} {\langle template2\rangle} {\langle template1\rangle}
```

---

Copies `\langle template1\rangle` of `\langle type\rangle` to a new name `\langle template2\rangle`: the copy can then be edited independent of teh original.

| Key-type               | Description of binding                                     |
|------------------------|------------------------------------------------------------|
|                        | Boolean variable, e.g. <code>\l_tmpa_bool</code>           |
| <code>choice</code>    | List of choice implementations (see Section 6)             |
| <code>commalist</code> | Comma list, e.g. <code>\l_tmpa_clist</code>                |
| <code>function</code>  | Function taking $N$ arguments, e.g. <code>\use_i:nn</code> |
| <code>instance</code>  |                                                            |
| <code>integer</code>   | Integer variable, e.g. <code>\l_tmpa_int</code>            |
| <code>length</code>    | Dimension variable, e.g. <code>\l_tmpa_dim</code>          |
| <code>muskip</code>    | Muskip variable, e.g. <code>\l_tmpa_muskip</code>          |
| <code>real</code>      | Floating-point variable, e.g. <code>\l_tmpa_fp</code>      |
| <code>skip</code>      | Skip variable, e.g. <code>\l_tmpa_skip</code>              |
| <code>tokenlist</code> | Token list variable, e.g. <code>\l_tmpa_t1</code>          |

Table 2: Bindings required for different key types when defining template implementations with `\DeclareTemplateCode`. Apart from `code`, `choice` and `function` all of these accept the key word `global` to carry out a global assignment.

## 6 Multiple choices

The `choice` key type implements multiple choice input. At the interface level, only the list of valid choices is needed:

```
\DeclareTemplateInterface { foo } { bar } { 0 }
{ key-name : choice { A, B, C } }
```

where the choices are given as a comma-list (which must therefore be wrapped in braces). A default value can also be given:

```
\DeclareTemplateInterface { foo } { bar } { 0 }
{ key-name : choice { A, B, C } = A }
```

At the implementation level, each choice is associated with code, using a nested key-value list.

```
\DeclareTemplateCode { foo } { bar } { 0 }
{
 key-name =
 {
 A = Code-A ,
 B = Code-B ,
 C = Code-C
 }
 ...
}
```

The two choice lists should match, but in the implementation a special `unknown` choice is also available. This can be used to ignore values and implement an “else” branch:

```
\DeclareTemplateCode { foo } { bar } { 0 }
{
 key-name =
```

```

{
 A = Code-A ,
 B = Code-B ,
 C = Code-C ,
 unknown = Else-code
}
}
{ ... }

```

The `unknown` entry must be the last one given, and should *not* be listed in the interface part of the template.

For keys which accept the values `true` and `false` both the boolean and choice key types can be used. As template interfaces are intended to prompt clarity at the design level, the boolean key type should be favoured, with the choice type reserved for keys which take arbitrary values.

## 7 Instances

After a template is defined it still needs to be put to use. The parameters that it expects need to be defined before it can be used in a document. Every time a template has parameters given to it, an *instance* is created, and this is the code that ends up in the document to perform the typesetting of whatever pieces of information are input into it.

For example, a template might say “here is a section with or without a number that might be centred or left aligned and print its contents in a certain font of a certain size, with a bit of a gap before and after it” whereas an instance declares “this is a section with a number, which is centred and set in 12 pt italic with a 10 pt skip before and a 12 pt skip after it”. Therefore, an instance is just a frozen version of a template with specific settings as chosen by the designer.

---

```
\DeclareInstance \DeclareInstance
 {<type>} {<instance>} {<template>} {<parameters>}
```

This function uses a `<template>` for an `<type>` to create an `<instance>`. The `<instance>` will be set up using the `<parameters>`, which will set some of the `<keys>` in the `<template>`.

As a practical example, consider a type for document sections (which might include chapters, parts, sections, *etc.*), which is called **sectioning**. One possible template for this type might be called `basic`, and one instance of this template would be a numbered section. The instance declaration might read:

```
\DeclareInstance { sectioning } { section-num } { basic }
{
 numbered = true ,
 justification = center ,
 font = \normalsize\itshape ,
 before-skip = 10pt ,
 after-skip = 12pt ,
}
```

Of course, the key names here are entirely imaginary, but illustrate the general idea of fixing some settings.

|                                   |                                                                                                                                                                                                                                            |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\IfInstanceExistsT</code>   | <code>\IfInstanceExistsTF {&lt;type&gt;} {&lt;instance&gt;} {&lt;true code&gt;} {&lt;false code&gt;}</code>                                                                                                                                |
| <code>\IfInstanceExistsF</code>   | Tests if the named <code>&lt;instance&gt;</code> of a <code>&lt;type&gt;</code> exists, and then inserts the appropriate code into the input stream.                                                                                       |
| <code>\DeclareInstanceCopy</code> | <code>\DeclareInstanceCopy {&lt;type&gt;} {&lt;instance2&gt;} {&lt;instance1&gt;}</code><br>Copies the <code>&lt;values&gt;</code> for <code>&lt;instance1&gt;</code> for an <code>&lt;type&gt;</code> to <code>&lt;instance2&gt;</code> . |

## 8 Document interface

After the instances have been chosen, document commands must be declared to use those instances in the document. `\UseInstance` calls instances directly, and this command should be used internally in document-level mark-up.

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\UseInstance</code> | <code>\UseInstance {&lt;type&gt;} {&lt;instance&gt;} {&lt;arguments&gt;}</code><br>Uses an <code>&lt;instance&gt;</code> of the <code>&lt;type&gt;</code> , which will require <code>&lt;arguments&gt;</code> as determined by the number specified for the <code>&lt;type&gt;</code> . The <code>&lt;instance&gt;</code> must have been declared before it can be used, otherwise an error is raised.                                                                                                                                                                                                             |
| <code>\UseTemplate</code> | <code>\UseTemplate {&lt;type&gt;} {&lt;template&gt;} {&lt;settings&gt;} {&lt;arguments&gt;}</code><br>Uses the <code>&lt;template&gt;</code> of the specified <code>&lt;type&gt;</code> , applying the <code>&lt;settings&gt;</code> and absorbing <code>&lt;arguments&gt;</code> as detailed by the <code>&lt;type&gt;</code> declaration. This in effect is the same as creating an instance using <code>\DeclareInstance</code> and immediately using it with <code>\UseInstance</code> , but without the instance having any further existence. It is therefore useful where a template needs to be used once. |

This function can also be used as the argument to `instance` key types:

```
\DeclareInstance { type } { template } { instance }
{
 instance-key =
 \UseTemplate { type2 } { template2 } { <settings> }
}
```

## 9 Changing existing definitions

Template parameters may be assigned specific defaults for instances to use if the instance declaration doesn't explicitly set those parameters. In some cases, the document designer will wish to edit these defaults to allow them to "cascade" to the instances. The alternative would be to set each parameter identically for each instance declaration, a tedious and error-prone process.

|                       |                                                                 |                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \EditTemplateDefaults | \EditTemplateDefaults<br>{<type>} {<template>} {<new defaults>} | Edits the <i>&lt;defaults&gt;</i> for a <i>&lt;template&gt;</i> for an <i>&lt;type&gt;</i> . The <i>&lt;new defaults&gt;</i> , given as a key–value list, replace the existing defaults for the <i>&lt;template&gt;</i> . This means that the change will apply to instances declared after the editing, but that instances which have already been created are unaffected. |
| \EditInstance         | \EditInstance<br>{<type>} {<instance>} {<new values>}           | Edits the <i>&lt;values&gt;</i> for an <i>&lt;instance&gt;</i> for an <i>&lt;type&gt;</i> . The <i>&lt;new values&gt;</i> , given as a key–value list, replace the existing values for the <i>&lt;instance&gt;</i> . This function is complementary to \EditTemplateDefaults: \EditInstance changes a single instance while leaving the template untouched.                 |

## 10 *Ad hoc* adjustment of templates

---

\SetTemplateKeys \SetTemplateKeys {<type>} {<template>} {<keyvals>}

At point of use it may be useful to apply changes to individual instances. This is supported as each template key is made available for adjustment using \SetTemplateKeys.

For example, after

```
\NewTypeType{MyObj}{0}
\DeclareTemplateInterface{MyObj}{TemplateA}{0}
{
 akey: tokenlist ,
 bkey: function{2}
}
\DeclareTemplateCode{MyObj}{TemplateA}{0}
{
 akey = SomeTokens ,
 bkey = \func:nn ,
}
```

the template keys could be adjusted in an *ad hoc* fashion using

```
\SetTemplateKeys{MyObj}{TemplateA}
{
 akey = OtherTokens ,
 bkey = \AltFunc:nn
}
```

## 11 Getting information about templates and instances

---

\ShowInstanceValues \ShowInstanceValues {<type>} {<instance>}

Shows the *<values>* for an *<instance>* of the given *<type>* at the terminal.

|                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <code>\ShowTemplateCode</code>                                                                                                                                                                                                                                                                                                                                                                                                      | <code>\ShowTemplateCode {&lt;type&gt;} {&lt;template&gt;}</code>      |
| Shows the <code>&lt;code&gt;</code> of a <code>&lt;template&gt;</code> for an <code>&lt;type&gt;</code> in the terminal.                                                                                                                                                                                                                                                                                                            |                                                                       |
| <code>\ShowTemplateDefaults</code>                                                                                                                                                                                                                                                                                                                                                                                                  | <code>\ShowTemplateDefaults {&lt;type&gt;} {&lt;template&gt;}</code>  |
| Shows the <code>&lt;default&gt;</code> values of a <code>&lt;template&gt;</code> for an <code>&lt;type&gt;</code> in the terminal.                                                                                                                                                                                                                                                                                                  |                                                                       |
| <code>\ShowTemplateInterface</code>                                                                                                                                                                                                                                                                                                                                                                                                 | <code>\ShowTemplateInterface {&lt;type&gt;} {&lt;template&gt;}</code> |
| Shows the <code>&lt;keys&gt;</code> and associated <code>&lt;key types&gt;</code> of a <code>&lt;template&gt;</code> for an <code>&lt;type&gt;</code> in the terminal.                                                                                                                                                                                                                                                              |                                                                       |
| <code>\ShowTemplateVariables</code>                                                                                                                                                                                                                                                                                                                                                                                                 | <code>\ShowTemplateVariables {&lt;type&gt;} {&lt;template&gt;}</code> |
| Shows the <code>&lt;variables&gt;</code> and associated <code>&lt;keys&gt;</code> of a <code>&lt;template&gt;</code> for an <code>&lt;type&gt;</code> in the terminal. Note that <code>code</code> and <code>choice</code> keys do not map directly to variables but to arbitrary code. For choice keys, each valid choice is shown as a separate entry in the list, with the key name and choice separated by a space, for example |                                                                       |

```
Template 'example' of type 'example' has variable mapping:
> demo unknown => \def \demo{?}
> demo c => \def \demo{c}
> demo b => \def \demo{b}
> demo a => \def \demo{a}.
```

would be shown for a choice key `demo` with valid choices `a`, `b` and `c`, plus code for an `unknown` branch.

## 12 The implementation

```

1 <@=@template>
2 <*2ekernel>
3 \message{templates,}
4 </2ekernel>
5 <*2ekernel | latexrelease>
6 \ExplSyntaxOn
7 <| latexrelease> \NewModuleRelease{2024/06/01}{lttemplates}
8 <| latexrelease> {Prototype~document~commands}%

```

## 12.1 Variables and constants

---

```
\c__template_code_root_tl
\c__template_defaults_root_tl
\c__template_instances_root_tl
\c__template_keytypes_root_tl
\c__template_key_order_root_tl
\c__template_restrict_root_tl
\c__template_values_root_tl
\c__template_vars_root_tl
```

---

So that literal values are kept to a minimum.

```
 9 \tl_const:Nn \c__template_code_root_tl { template~code~>~ }
10 \tl_const:Nn \c__template_defaults_root_tl { template~defaults~>~ }
11 \tl_const:Nn \c__template_instances_root_tl { template~instance~>~ }
12 \tl_const:Nn \c__template_keytypes_root_tl { template~key~types~>~ }
13 \tl_const:Nn \c__template_key_order_root_tl { template~key~order~>~ }
14 \tl_const:Nn \c__template_values_root_tl { template~values~>~ }
15 \tl_const:Nn \c__template_vars_root_tl { template~vars~>~ }
```

---

```
\c__template_keytypes_arg_seq
```

---

A list of keytypes which also need additional data (an argument), used to parse the keytype correctly.

```
16 \seq_const_from_clist:Nn \c__template_keytypes_arg_seq
17 { choice , function , instance }
```

---

```
\g__template_type_prop
```

 For storing types and the associated number of arguments.

```
18 \prop_new:N \g__template_type_prop
```

---

```
\l__template_assignments_tl
```

---

When creating an instance, the assigned values are collected here.

```
19 \tl_new:N \l__template_assignments_tl
```

---

```
\l__template_default_tl
```

 The default value for a key is recovered here from the property list in which it is stored.

```
20 \tl_new:N \l__template_default_tl
```

---

```
\l__template_error_bool
```

 A flag for errors to be carried forward.

```
21 \bool_new:N \l__template_error_bool
```

---

```
\l__template_global_bool
```

 Used to indicate that assignments should be global.

```
22 \bool_new:N \l__template_global_bool
```

---

```
\l__template_key_name_tl
\l__template_keytype_tl
\l__template_keytype_arg_tl
\l__template_value_tl
\l__template_var_tl
```

---

When defining each key in a template, the name and type of the key need to be separated and stored. Any argument needed by the keytype is also stored separately.

```
23 \tl_new:N \l__template_key_name_tl
24 \tl_new:N \l__template_keytype_tl
25 \tl_new:N \l__template_keytype_arg_tl
26 \tl_new:N \l__template_value_tl
27 \tl_new:N \l__template_var_tl
```

---

```
\l__template_keytypes_prop
```

To avoid needing too many difficult-to-follow csname assignments, various scratch token registers are used to build up data, which is then transferred

```
\l__template_key_order_seq
\l__template_values_prop
\l__template_vars_prop
```

---

```
28 \prop_new:N \l__template_keytypes_prop
29 \seq_new:N \l__template_key_order_seq
30 \prop_new:N \l__template_values_prop
31 \prop_new:N \l__template_vars_prop
```

---

```
\l__template_tmp_clist
```

Scratch space.

```
\l__template_tmp_dim
\l__template_tmp_int
\l__template_tmp_muskip
\l__template_tmp_skip
\l__template_tmp_tl
```

---

```
32 \clist_new:N \l__template_tmp_clist
33 \dim_new:N \l__template_tmp_dim
34 \int_new:N \l__template_tmp_int
35 \muskip_new:N \l__template_tmp_muskip
36 \skip_new:N \l__template_tmp_skip
37 \tl_new:N \l__template_tmp_tl
```

---

```
\s__template_mark
```

Internal scan marks.

---

```
\s__template_stop
```

---

```
38 \scan_new:N \s__template_mark
39 \scan_new:N \s__template_stop
```

---

```
\q__template_nil
```

Internal quarks.

---

```
40 \quark_new:N \q__template_nil
```

```
__template_quark_if_nil_p:n
```

Branching quark conditional.

```
__template_quark_if_nil:nTF
```

---

```
41 __kernel_quark_new_conditional:Nn __template_quark_if_nil:N { F }
```

*(End of definition for \\_\\_template\_quark\_if\_nil:nTF.)*

## 12.2 Testing existence and validity

There are a number of checks needed for either the existence of a type, template or instance. There are also some for the validity of a particular call. All of these are collected up here.

`\_\_template\_execute\_if\_arg\_agree:nmT`

A test agreement between the number of arguments for the template type and that specified when creating a template. This is not done as a separate conditional for efficiency and better error message

```

42 \cs_new_protected:Npn __template_execute_if_arg_agree:nmT #1#2#3
43 {
44 \prop_get:NnN \g___template_type_prop {#1} \l___template_tmp_tl
45 \int_compare:nNnTF {#2} = \l___template_tmp_tl
46 {#3}
47 {
48 \msg_error:nneee { template } { argument-number-mismatch }
49 {#1} { \l___template_tmp_tl } {#2}
50 }
51 }

```

(End of definition for `\_\_template\_execute\_if\_arg\_agree:nmT`.)

`\_\_template\_execute\_if\_code\_exist:nnT`

A template is only fully declared if the code has been set up, which can be checked by looking for the template function itself.

```

52 \cs_new_protected:Npn __template_execute_if_code_exist:nnT #1#2#3
53 {
54 \cs_if_exist:cTF { \c___template_code_root_tl #1 / #2 }
55 {#3}
56 { \msg_error:nnnn { template } { no-template-code } {#1} {#2} }
57 }

```

(End of definition for `\_\_template\_execute\_if\_code\_exist:nnT`.)

`\_\_template\_execute\_if\_keytype\_exist:nT`  
`\_\_template\_execute\_if\_keytype\_exist:VT`

The test for valid keytypes looks for a function to set up the key, which is part of the “code” side of the template definition. This avoids having different lists for the two parts of the process.

```

58 \cs_new_protected:Npn __template_execute_if_keytype_exist:nT #1#2
59 {
60 \cs_if_exist:cTF { \c___template_store_value_ #1 :n }
61 {#2}
62 { \msg_error:nnn { template } { unknown-keytype } {#1} }
63 }
64 \cs_generate_variant:Nn __template_execute_if_keytype_exist:nT { V }

```

(End of definition for `\_\_template\_execute\_if\_keytype\_exist:nT`.)

`\_\_template\_execute\_if\_type\_exist:nT`

To check that a particular type is valid.

```

65 \cs_new_protected:Npn __template_execute_if_type_exist:nT #1#2
66 {
67 \prop_if_in:NnTF \g___template_type_prop {#1}
68 {#2}
69 { \msg_error:nnn { template } { unknown-type } {#1} }
70 }

```

(End of definition for `\_\_template\_execute\_if\_type\_exist:nT`.)

```

__template_execute_if_keys_exist:nnt
To check that the keys for a template have been set up before trying to create any code,
a simple check for the correctly-named keytype property list.

71 \cs_new_protected:Npn __template_if_keys_exist:nnt #1#2#3
72 {
73 \cs_if_exist:cTF { \c__template_keytypes_root_tl #1 / #2 }
74 {#3}
75 { \msg_error:nnnn { template } { unknown-template } {#1} {#2} }
76 }

(End of definition for __template_execute_if_keys_exist:nnt.)

__template_if_key_value:nTF
__template_if_key_value:VTF
Tests for the first token in a string being \KeyValue.

77 \prg_new_conditional:Npnn __template_if_key_value:n #1 { T , F , TF }
78 {
79 \str_if_eq:noTF { \KeyValue } { \tl_head:w #1 \q_nil \q_stop }
80 \prg_return_true:
81 \prg_return_false:
82 }
83 \prg_generate_conditional_variant:Nnn __template_if_key_value:n { V } { T , F , TF }

(End of definition for __template_if_key_value:nTF.)

__template_if_instance_exist:nnTF
Testing for an instance

84 \prg_new_conditional:Npnn __template_if_instance_exist:nn #1#2 { T , F , TF }
85 {
86 \cs_if_exist:cTF { \c__template_instances_root_tl #1 / #2 }
87 \prg_return_true:
88 \prg_return_false:
89 }

(End of definition for __template_if_instance_exist:nnTF.)

__template_if_use_template:nTF
Tests for the first token in a string being \UseTemplate.

90 \prg_new_conditional:Npnn __template_if_use_template:n #1 { TF }
91 {
92 \str_if_eq:noTF { \UseTemplate } { \tl_head:w #1 \q_nil \q_stop }
93 \prg_return_true:
94 \prg_return_false:
95 }

(End of definition for __template_if_use_template:nTF.)

```

### 12.3 Saving and recovering property lists

The various property lists for templates have to be shuffled in and out of storage.

The defaults and keytypes are transferred from the scratch property lists to the “proper” lists for the template being created.

```

96 \cs_new_protected:Npn __template_store_defaults:nn
97 {
98 \debug_suspend:
99 \prop_gclear_new:c { \c__template_defaults_root_tl #1 / #2 }
100 \prop_gset_eq:cN { \c__template_defaults_root_tl #1 / #2 }
101 \l__template_values_prop

```

```

102 \debug_resume:
103 }
104 \cs_new_protected:Npn __template_store_keytypes:nn #1#2
105 {
106 \debug_suspend:
107 \prop_if_exist:cTF { \c_template_keytypes_root_tl #1 / #2 }
108 {
109 \msg_info:nnnn { template } { declare-template-interface } {#1} {#2}
110 \prop_gclear:c { \c_template_keytypes_root_tl #1 / #2 }
111 }
112 { \prop_new:c { \c_template_keytypes_root_tl #1 / #2 } }
113 \prop_gset_eq:cN { \c_template_keytypes_root_tl #1 / #2 }
114 \l_template_keytypes_prop
115 \seq_gclear_new:c { \c_template_key_order_root_tl #1 / #2 }
116 \seq_gset_eq:cN { \c_template_key_order_root_tl #1 / #2 }
117 \l_template_key_order_seq
118 \debug_resume:
119 }
120 \cs_new_protected:Npn __template_store_values:nn #1#2
121 {
122 \debug_suspend:
123 \prop_clear_new:c { \c_template_values_root_tl #1 / #2 }
124 \prop_set_eq:cN { \c_template_values_root_tl #1 / #2 }
125 \l_template_values_prop
126 \debug_resume:
127 }
128 \cs_new_protected:Npn __template_store_vars:nn #1#2
129 {
130 \debug_suspend:
131 \prop_gclear_new:c { \c_template_vars_root_tl #1 / #2 }
132 \prop_gset_eq:cN { \c_template_vars_root_tl #1 / #2 }
133 \l_template_vars_prop
134 \debug_resume:
135 }

```

(End of definition for `\__template_store_defaults:nn` and others.)

`\__template_recover_defaults:nn`  
`\__template_recover_keytypes:nn`  
`\__template_recover_values:nn`

Recovering the stored data for a template is rather less complex than storing it. All that happens is the data is transferred from the permanent to the scratch storage. However, we need to check the scratch storage does exist.

```

136 \cs_new_protected:Npn __template_recover_defaults:nn #1#2
137 {
138 \prop_if_exist:cTF
139 { \c_template_defaults_root_tl #1 / #2 }
140 {
141 \prop_set_eq:Nc \l_template_values_prop
142 { \c_template_defaults_root_tl #1 / #2 }
143 }
144 { \prop_clear:N \l_template_values_prop }
145 }
146 \cs_new_protected:Npn __template_recover_keytypes:nn #1#2
147 {
148 \prop_if_exist:cTF
149 { \c_template_keytypes_root_tl #1 / #2 }

```

```

150 {
151 \prop_set_eq:Nc \l__template_keytypes_prop
152 { \c__template_keytypes_root_tl #1 / #2 }
153 }
154 { \prop_clear:N \l__template_keytypes_prop }
155 \seq_if_exist:cTF { \c__template_key_order_root_tl #1 / #2 }
156 {
157 \seq_set_eq:Nc \l__template_key_order_seq
158 { \c__template_key_order_root_tl #1 / #2 }
159 }
160 { \seq_clear:N \l__template_key_order_seq }
161 }
162 \cs_new_protected:Npn __template_recover_values:nn #1#2
163 {
164 \prop_if_exist:cTF
165 { \c__template_values_root_tl #1 / #2 }
166 {
167 \prop_set_eq:Nc \l__template_values_prop
168 { \c__template_values_root_tl #1 / #2 }
169 }
170 { \prop_clear:N \l__template_values_prop }
171 }
172 \cs_new_protected:Npn __template_recover_vars:nn #1#2
173 {
174 \prop_if_exist:cTF
175 { \c__template_vars_root_tl #1 / #2 }
176 {
177 \prop_set_eq:Nc \l__template_vars_prop
178 { \c__template_vars_root_tl #1 / #2 }
179 }
180 { \prop_clear:N \l__template_vars_prop }
181 }

```

(End of definition for `\__template_recover_defaults:nn` and others.)

## 12.4 Creating new template types

`\__template_define_type:nn`  
`\__template_declare_type:nn`

Although the type is the “top level” of the template system, it is actually very easy to implement. All that happens is that the number of arguments required is recorded, indexed by the name of the type.

```

182 \cs_new_protected:Npn __template_define_type:nn #1#2
183 {
184 \prop_if_in:NnTF \g__template_type_prop {#1}
185 { \msg_error:nnn { template } { type-already-defined } {#1} }
186 { __template_declare_type:nn {#1} {#2} }
187 }
188 \cs_new_protected:Npn __template_declare_type:nn #1#2
189 {
190 \int_set:Nn \l__template_tmp_int {#2}
191 \int_compare:nTF { 0 <= \l__template_tmp_int <= 9 }
192 {
193 \msg_info:nnnV { template } { declare-type }
194 {#1} \l__template_tmp_int
195 \prop_gput:NnV \g__template_type_prop {#1}

```

```

196 \l__template_tmp_int
197 }
198 {
199 \msg_error:n{template } { bad-number-of-arguments }
200 {#1} \l__template_tmp_int
201 }
202 }
```

*(End of definition for \\_\\_template\\_define\\_type:nn and \\_\\_template\\_declare\\_type:nn.)*

## 12.5 Design part of template declaration

The “design” part of a template declaration defines the general behaviour of each key, and possibly a default value. However, it does not include the implementation. This means that what happens here is the two properties are saved to appropriate lists, which can then be used later to recover the information when implementing the keys.

The main function for the “design” part of creating a template starts by checking that the type exists and that the number of arguments required agree. If that is all fine, then the two storage areas for defaults and keytypes are initialised. The mechanism is then set up for the l3keys module to actually parse the keys. Finally, the code hands off to the storage routine to save the parsed information properly.

```

203 \cs_new_protected:Npn __template_declare_template_keys:nnnn #1#2#3#4
204 {
205 __template_execute_if_type_exist:nT {#1}
206 {
207 __template_execute_if_arg_agree:nnT {#1} {#3}
208 {
209 \prop_clear:N \l__template_values_prop
210 \prop_clear:N \l__template_keytypes_prop
211 \seq_clear:N \l__template_key_order_seq
212 \keyval_parse>NNn
213 __template_parse_keys_elt:n __template_parse_keys_elt:nn {#4}
214 __template_store_defaults:nn {#1} {#2}
215 __template_store_keytypes:nn {#1} {#2}
216 }
217 }
218 }
```

*(End of definition for \\_\\_template\\_declare\\_template\\_keys:nnnn.)*

Processing the key part of the key–value pair is always carried out using this function, even if a value was found. First, the key name is separated from the keytype, and if necessary the keytype is separated into two parts. This information is then used to check that the keytype is valid, before storing the keytype (plus argument if necessary) as a property of the key name. The key name is also stored (in braces) in the token list to record the order the keys are defined in.

```

219 \cs_new_protected:Npn __template_parse_keys_elt:n #1
220 {
221 __template_split_keytype:n {#1}
222 \bool_if:NF \l__template_error_bool
223 {
224 __template_execute_if_keytype_exist:VT \l__template_keytype_tl
```

```

225 {
226 \seq_map_function:NN \c__template_keytypes_arg_seq
227 __template_parse_keys_elt_aux:n
228 \bool_if:NF \l__template_error_bool
229 {
230 \seq_if_in:NoTF \l__template_key_order_seq
231 \l__template_key_name_tl
232 {
233 \msg_error:nnV { template } { duplicate-key-interface }
234 \l__template_key_name_tl
235 }
236 { __template_parse_keys_elt_aux: }
237 }
238 }
239 }
240 }
241 \cs_new_protected:Npn __template_parse_keys_elt_aux:n #1
242 {
243 \str_if_eq:VnT \l__template_keytype_tl {#1}
244 {
245 \tl_if_empty:NT \l__template_keytype_arg_tl
246 {
247 \msg_error:nnn { template } { keytype-requires-argument } {#1}
248 \bool_set_true:N \l__template_error_bool
249 \seq_map_break:
250 }
251 }
252 }
253 \cs_new_protected:Npn __template_parse_keys_elt_aux:
254 {
255 \tl_set:Ne \l__template_tmp_tl
256 {
257 \l__template_keytype_tl
258 \tl_if_empty:NF \l__template_keytype_arg_tl
259 { { \l__template_keytype_arg_tl } }
260 }
261 \prop_put:NVV \l__template_keytypes_prop \l__template_key_name_tl
262 \l__template_tmp_tl
263 \seq_put_right:NV \l__template_key_order_seq \l__template_key_name_tl
264 \str_if_eq:VnT \l__template_keytype_tl { choice }
265 {
266 \clist_if_in:NnT \l__template_keytype_arg_tl { unknown }
267 { \msg_error:nn { template } { choice-unknown-reserved } }
268 }
269 }

```

(End of definition for `\__template_parse_keys_elt:n`, `\__template_parse_keys_elt_aux:n`, and `\__template_parse_keys_elt_aux:..`)

`\__template_parse_keys_elt:nn` For keys which have a default, the keytype and key name are first separated out by the `\__template_parse_keys_elt:n` routine, before storing the default value in the scratch property list.

```

270 \cs_new_protected:Npn __template_parse_keys_elt:nn #1#2
271 {

```

```

272 __template_parse_keys_elt:n {#1}
273 \use:c { __template_store_value_ \l__template_keytype_tl :n } {#2}
274 }

```

(End of definition for `\__template_parse_keys_elt:nn`.)

`\__template_split_keytype:n  
  \__template_split_keytype_aux:w`

The keytype and key name should be separated by `:`. As the definition might be given inside or outside of a code block, the category code of colons is standardised. After that, the standard delimited argument method is used to separate the two parts.

```

275 \cs_new_protected:Npe __template_split_keytype:n #1
276 {
277 \exp_not:N \bool_set_false:N \exp_not:N \l__template_error_bool
278 \tl_set:Nn \exp_not:N \l__template_tmp_tl {#1}
279 \tl_replace_all:Nnn \exp_not:N \l__template_tmp_tl { : } { \token_to_str:N : }
280 \tl_if_in:VnTF \exp_not:N \l__template_tmp_tl { \token_to_str:N : }
281 {
282 \exp_not:n
283 {
284 \tl_clear:N \l__template_key_name_tl
285 \exp_after:wN __template_split_keytype_aux:w
286 \l__template_tmp_tl \s__template_stop
287 }
288 }
289 {
290 \exp_not:N \bool_set_true:N \exp_not:N \l__template_error_bool
291 \msg_error:nnn { template } { missing-keytype } {#1}
292 }
293 }
294 \use:e
295 {
296 \cs_new_protected:Npn \exp_not:N __template_split_keytype_aux:w
297 #1 \token_to_str:N : #2 \s__template_stop
298 {
299 \tl_put_right:Ne \exp_not:N \l__template_key_name_tl
300 {
301 \exp_not:N \tl_trim_spaces:e
302 { \exp_not:N \tl_to_str:n {#1} }
303 }
304 \tl_if_in:nnTF {#2} { \token_to_str:N : }
305 {
306 \tl_put_right:Nn \exp_not:N \l__template_key_name_tl
307 { \token_to_str:N : }
308 \exp_not:N __template_split_keytype_aux:w #2 \s__template_stop
309 }
310 }
311 \exp_not:N \tl_if_empty:NTF \exp_not:N \l__template_key_name_tl
312 {
313 \msg_error:nnn { template } { empty-key-name }
314 { \token_to_str:N : #2 }
315 }
316 { \exp_not:N __template_split_keytype_arg:n {#2} }
317 }
318 }
319 }

```

(End of definition for `\_template_split_keytype:n` and `\_template_split_keytype_aux:w`.)

```
_template_split_keytype_arg:n
_template_split_keytype_arg:v
_template_split_keytype_arg_aux:n
_template_split_keytype_arg_aux:w
```

The second stage of sorting out the keytype is to check for an argument. As there is no convenient delimiting token to look for, a check is made instead for each possible text value for the keytype. To keep things faster, this only involves the keytypes that need an argument. If a match is made, then a check is also needed to see that it is at the start of the keytype information. All being well, the split can then be applied. Any non-matching keytypes are assumed to be “correct” as given, and are left alone (this is checked by other code).

```
320 \cs_new_protected:Npn _template_split_keytype_arg:n #1
321 {
322 \tl_set:Ne \l__template_keytype_tl { \tl_trim_spaces:n {#1} }
323 \tl_clear:N \l__template_keytype_arg_tl
324 \cs_set_protected:Npn _template_split_keytype_arg_aux:n ##1
325 {
326 \tl_if_in:nnT {#1} {##1}
327 {
328 \cs_set:Npn _template_split_keytype_arg_aux:w
329 #####1 ##1 #####2 \s__template_stop
330 {
331 \tl_if_blank:nT {#####1}
332 {
333 \tl_set:Ne \l__template_keytype_tl
334 { \tl_trim_spaces:n {##1} }
335 \tl_if_blank:nF {#####2}
336 {
337 \tl_set:Ne \l__template_keytype_arg_tl
338 { \use:n #####2 }
339 }
340 \seq_map_break:
341 }
342 }
343 _template_split_keytype_arg_aux:w #1 \s__template_stop
344 }
345 }
346 \seq_map_function:NN \c__template_keytypes_arg_seq
347 _template_split_keytype_arg_aux:n
348 }
349 \cs_generate_variant:Nn _template_split_keytype_arg:n { V }
350 \cs_new:Npn _template_split_keytype_arg_aux:n #1 { }
351 \cs_new:Npn _template_split_keytype_arg_aux:w #1 \s__template_stop { }
```

(End of definition for `\_template_split_keytype_arg:n`, `\_template_split_keytype_arg_aux:n`, and `\_template_split_keytype_arg_aux:w`.)

### 12.5.1 Storing values

As `lttemplates` pre-processes key values for efficiency reasons, there is a need to convert the values given as defaults into “ready to use” data. The same general idea is true when an instance is declared. However, assignments are not made until an instance is used, and so there has to be some intermediate storage. Furthermore, the ability to delay evaluation of results is needed. To achieve these aims, a series of “process and store” functions are defined here.

All of the information about the key (the key name and the keytype) is already stored as variables. The same property list is always used to store the data, meaning that the only argument required is the value to be processed and potentially stored.

```
__template_store_value_boolean:n
352 \cs_new_protected:Npn __template_store_value_boolean:n #1
353 { \prop_put:Non \l__template_values_prop \l__template_key_name_tl {#1} }

(End of definition for __template_store_value_boolean:n.)
```

With no need to worry about delayed evaluation, these keytypes all just store the input directly.

```
354 \cs_new_protected:Npn __template_store_value:n #1
355 { \prop_put:Non \l__template_values_prop \l__template_key_name_tl {#1} }
356 \cs_new_eq:NN __template_store_value_choice:n __template_store_value:n
357 \cs_new_eq:NN __template_store_value_function:n __template_store_value:n
358 \cs_new_eq:NN __template_store_value_instance:n __template_store_value:n

(End of definition for __template_store_value:n and others.)
```

Storing values in `\l_\_template\_values\_prop` is in most cases the same.

```
359 \cs_new_protected:Npn __template_store_value_aux:Nn #1#2
360 { \prop_put:Non \l__template_values_prop \l__template_key_name_tl {#2} }
361 \cs_new_protected:Npn __template_store_value_integer:n
362 { __template_store_value_aux:Nn \int_eval:n }
363 \cs_new_protected:Npn __template_store_value_length:n
364 { __template_store_value_aux:Nn \dim_eval:n }
365 \cs_new_protected:Npn __template_store_value_muskip:n
366 { __template_store_value_aux:Nn \muskip_eval:n }
367 \cs_new_protected:Npn __template_store_value_real:n
368 { __template_store_value_aux:Nn \fp_eval:n }
369 \cs_new_protected:Npn __template_store_value_skip:n
370 { __template_store_value_aux:Nn \skip_eval:n }
371 \cs_new_protected:Npn __template_store_value_tokenlist:n
372 { __template_store_value_aux:Nn \use:n }
373 \cs_new_eq:NN __template_store_value_commalist:n __template_store_value_tokenlist:n

(End of definition for __template_store_value_aux:Nn and others.)
```

## 12.6 Implementation part of template declaration

The main function for implementing a template starts with a couple of simple checks to make sure that there are no obvious mistakes: the number of arguments must agree and the template keys must have been declared.

```
374 \cs_new_protected:Npn __template_declare_template_code:nnnnn #1#2#3#4#5
375 {
376 __template_execute_if_type_exist:nT {#1}
377 {
378 __template_execute_if_arg_agree:nnT {#1} {#3}
379 {
380 __template_if_keys_exist:nnT {#1} {#2}
381 {
382 __template_store_key_implementation:nnn {#1} {#2} {#4}
383 \regex_match:nnTF { \c { AssignTemplateKeys } } {#5}
```

```

384 { __template_declare_template_code:nnnn {#1} {#2} {#3} {#5} }
385 {
386 __template_declare_template_code:nnnn
387 {#1} {#2} {#3} { \AssignTemplateKeys #5 }
388 }
389 }
390 }
391 }
392 }
393 \cs_new_protected:Npn __template_declare_template_code:nnnn #1#2#3#4
394 {
395 \cs_if_exist:cT { \c__template_code_root_tl #1 / #2 }
396 { \msg_info:nnnn { template } { declare-template-code } {#1} {#2} }
397 \cs_generate_from_arg_count:cNnn
398 { \c__template_code_root_tl #1 / #2 }
399 \cs_gset_protected:Npn {#3} {#4}
400 }

```

*(End of definition for \\_\_template\_declare\_template\_code:nnnn and  
\\_\_template\_declare\_template\_code:nnnn.)*

\\_\_template\_store\_key\_implementation:nnn

Actually storing the implementation part of a template is quite easy as it only requires the list of keys given to be turned into a property list. There is also some error-checking to do, hence the need to have the list of defined keytypes available. In certain cases (when choices are involved) parsing the key results in changes to the default values. That is why they are loaded and then saved again.

```

401 \cs_new_protected:Npn __template_store_key_implementation:nnn #1#2#3
402 {
403 __template_recover_defaults:nn {#1} {#2}
404 __template_recover_keytypes:nn {#1} {#2}
405 \prop_clear:N \l__template_vars_prop
406 \keyval_parse:nnn
407 { __template_parse_vars_elt:n } { __template_parse_vars_elt:nnn { #1 / #2 } } {#3}
408 __template_store_vars:nn {#1} {#2}
409 \prop_map_inline:Nn \l__template_keytypes_prop
410 { \msg_error:nnnn { template } { key-not-implemented } {##1} {#2} {#1} }
411 }

```

*(End of definition for \\_\_template\_store\_key\_implementation:nnn.)*

\\_\_template\_parse\_vars\_elt:n

At the implementation stage, every key must have a value given. So this is an error function.

```

412 \cs_new_protected:Npn __template_parse_vars_elt:n #1
413 { \msg_error:nnn { template } { key-no-variable } {#1} }

```

*(End of definition for \\_\_template\_parse\_vars\_elt:n.)*

\\_\_template\_parse\_vars\_elt:nnn  
\\_\_template\_parse\_vars\_elt\_aux:nn  
\\_\_template\_parse\_vars\_elt\_aux:nnw  
\\_\_template\_parse\_vars\_elt\_aux:nnn  
\\_\_template\_parse\_vars\_elt\_aux:nne  
\\_\_template\_parse\_vars\_elt\_key:nn

The actual storage part here is very simple: the storage bin name is placed into the property list. At the same time, a comparison is made with the keytypes defined earlier: if there is a mismatch then an error is raised.

```

414 \cs_new_protected:Npn __template_parse_vars_elt:nnn #1#2#3
415 {
416 \tl_set:Ne \l__template_key_name_tl
417 { \tl_trim_spaces:e { \tl_to_str:n {#2} } }

```

```

418 \prop_get:NVNTF \l__template_keytypes_prop
419 \l__template_key_name_tl
420 \l__template_keytype_tl
421 {
422 __template_split_keytype_arg:V \l__template_keytype_tl
423 __template_parse_vars_elt_aux:nn {#1} {#3}
424 \prop_remove:NV \l__template_keytypes_prop \l__template_key_name_tl
425 }
426 { \msg_error:n { template } { unknown-key } {#2} }
427 }

Split off any leading global and they look for the way to implement.
428 \cs_new_protected:Npn __template_parse_vars_elt_aux:nn #1#2
429 {
430 __template_parse_vars_elt_aux:nw {#1} #2 global global \s__template_stop
431 }
432 \cs_new_protected:Npn __template_parse_vars_elt_aux:nw
433 #1#2 global #3 global #4 \s__template_stop
434 {
435 \tl_if_blank:nTF {#4}
436 { __template_parse_vars_elt_aux:nnn {#1} { } {#2} }
437 {
438 \tl_if_blank:nTF {#2}
439 {
440 __template_parse_vars_elt_aux:nne
441 {#1} { global } { \tl_trim_spaces:n {#3} }
442 }
443 { \msg_error:n { template } { bad-variable } { #2 global #3 } }
444 }
445 }
446 \cs_new_protected:Npn __template_parse_vars_elt_aux:nnn #1#2#3
447 {
448 \str_case:VnF \l__template_keytype_tl
449 {
450 { choice } { __template_implement_choices:nn {#1} {#3} }
451 { function }
452 {
453 \cs_if_exist:NF #3
454 { \cs_new:Npn #3 { } }
455 __template_parse_vars_elt_key:nn {#1}
456 {
457 .code:n =
458 {
459 \cs_generate_from_arg_count>NNnn
460 \exp_not:N #3
461 \exp_not:c
462 { cs_\str_if_eq:nnT {#1} { global } { g } set:Npn }
463 { \exp_not:V \l__template_keytype_arg_tl }
464 {##1}
465 }
466 }
467 \prop_put:NVn \l__template_vars_prop
468 \l__template_key_name_tl {#2#3}
469 }
470 { instance }

```

```

471 {
472 __template_parse_vars_elt_key:nn {#1}
473 {
474 .code:n =
475 {
476 \exp_not:c
477 { cs_ \str_if_eq:nnT {#1} { global } { g } set:Npn }
478 \exp_not:N #3 { \UseInstance {##1} }
479 }
480 }
481 \prop_put:NVn \l__template_vars_prop
482 \l__template_key_name_tl {#2#3}
483 }
484 }
485 {
486 \tl_if_single:nTF {#3}
487 {
488 \cs_if_exist:NF #3
489 { \use:c { __template_map_var_type: _new:N } #3 }
490 __template_parse_vars_elt_key:nn {#1}
491 {
492 . __template_map_var_type:
493 - \str_if_eq:nnT {#1} { global } { g } set:N
494 = \exp_not:N #3
495 }
496 \prop_put:NVn \l__template_vars_prop
497 \l__template_key_name_tl {#2#3}
498 }
499 { \msg_error:nnn { template } { bad-variable } { #2#3 } }
500 }
501 }
502 \cs_generate_variant:Nn __template_parse_vars_elt_aux:nnn { nne }
503 \cs_new_protected:Npn __template_parse_vars_elt_key:nn #1#2
504 {
505 \keys_define:ne { template / #1 }
506 { \l__template_key_name_tl #2 }
507 }

```

(End of definition for `\__template_parse_vars_elt:nnn` and others.)

`\__template_map_var_type:` Turn a “friendly” variable type into an `expl3` one.

```

508 \cs_new:Npn __template_map_var_type:
509 {
510 \str_case:Vn \l__template_keytype_tl
511 {
512 { boolean } { bool }
513 { commalist } {clist}
514 { integer } { int }
515 { length } { dim }
516 { muskip } { muskip }
517 { real } { fp }
518 { skip } { skip }
519 { tokenlist } { tl }
520 }
521 }

```

(End of definition for `\_template_map_var_type::`)

Implementing choices requires a second key–value loop. So after a little set-up, the standard parser is called.

```
522 \cs_new_protected:Npn _template_implement_choices:nn #1#2
523 {
524 \clist_set:NV \l__template_tmp_clist \l__template_keytype_arg_tl
525 \prop_put:NVn \l__template_vars_prop \l__template_key_name_tl { }
526 \keys_define:ne { template / #1 } { \l__template_key_name_tl .choice: }
527 \keyval_parse:nnn
528 { _template_implement_choice_elt:n }
529 { _template_implement_choice_elt:nnn {#1} }
530 {#2}
531 \prop_get:NVNT \l__template_values_prop \l__template_key_name_tl
532 \l__template_tmp_tl
533 { _template_implement_choices_default: }
534 \clist_if_empty:NF \l__template_tmp_clist
535 {
536 \clist_map_inline:Nn \l__template_tmp_clist
537 { \msg_error:nnn { template } { choice-not-implemented } {##1} }
538 }
539 }
```

A sanity check for the default value, so that an error is raised now and not when converting to assignments.

```
540 \cs_new_protected:Npn _template_implement_choices_default:
541 {
542 \tl_set:Ne \l__template_tmp_tl
543 { \l__template_key_name_tl \c_space_tl \l__template_tmp_tl }
544 \prop_if_in:NVF \l__template_vars_prop \l__template_tmp_tl
545 {
546 \tl_set:Ne \l__template_tmp_tl
547 { \l__template_key_name_tl \c_space_tl \l__template_tmp_tl }
548 \prop_if_in:NVF \l__template_vars_prop \l__template_tmp_tl
549 {
550 \prop_get:NVN \l__template_keytypes_prop \l__template_key_name_tl
551 \l__template_tmp_tl
552 \l__template_split_keytype_arg:V \l__template_tmp_tl
553 \prop_get:NVN \l__template_values_prop \l__template_key_name_tl
554 \l__template_tmp_tl
555 \msg_error:nnVV { template } { unknown-default-choice }
556 \l__template_key_name_tl
557 \l__template_key_name_tl
558 }
559 }
560 }
```

(End of definition for `\_template_implement_choices:nn` and  
`\_template_implement_choices_default::`)

The actual storage of the implementation of a choice is mainly about error checking. The code here ensures that all choices have to have been declared, apart from the special `unknown` choice, which must come last. The code for each choice is stored along with the key name in the variables property list.

```

561 \cs_new_protected:Npn __template_implement_choice_elt:nnn #1#2#3
562 {
563 \clist_if_empty:NTF \l__template_tmp_clist
564 {
565 \str_if_eq:nnTF {#2} { unknown }
566 { __template_implement_choice_elt_aux:nnn {#1} {#2} {#3} }
567 { __template_implement_choice_elt_aux:n {#2} }
568 }
569 {
570 \clist_if_in:NnTF \l__template_tmp_clist {#2}
571 {
572 \clist_remove_all:Nn \l__template_tmp_clist {#2}
573 __template_implement_choice_elt_aux:nnn {#1} {#2} {#3}
574 }
575 { __template_implement_choice_elt_aux:n {#2} }
576 }
577 }
578 \cs_new_protected:Npn __template_implement_choice_elt_aux:n #1
579 {
580 \prop_get:NVN \l__template_keytypes_prop \l__template_key_name_tl
581 \l__template_tmp_tl
582 __template_split_keytype_arg:V \l__template_tmp_tl
583 \msg_error:nnVn { template } { unknown-choice } \l__template_key_name_tl {#1}
584 }
585 \cs_new_protected:Npn __template_implement_choice_elt_aux:nnn #1#2#3
586 {
587 \keys_define:ne { template / #1 }
588 { \l__template_key_name_tl / #2 .code:n = { \exp_not:n {#3} } }
589 \tl_set:Ne \l__template_tmp_tl
590 { \l__template_key_name_tl \c_space_tl #2 }
591 \prop_put:NVn \l__template_vars_prop \l__template_tmp_tl {#3}
592 }
593 \cs_new_protected:Npn __template_implement_choice_elt:n #1
594 {
595 \msg_error:nnVn { template } { choice-requires-code }
596 \l__template_key_name_tl {#1}
597 }

```

(End of definition for `\__template_implement_choice_elt:nnn` and others.)

## 12.7 Editing template defaults

`\__template_edit_defaults:nnn`

Editing the template defaults means getting the values back out of the store, then parsing the list of new values before putting the updated list back into storage.

```

598 \cs_new_protected:Npn __template_edit_defaults:nnn #1#2#3
599 {
600 __template_if_keys_exist:nnT {#1} {#2}
601 {
602 __template_recover_defaults:nn {#1} {#2}
603 __template_parse_values:nnn {#1} {#2} {#3}
604 __template_store_defaults:nn {#1} {#2}
605 }
606 }

```

(End of definition for `\_template_edit_defaults:nnn`.)

`\_template_parse_values:nnn` The routine to parse values is the same for both editing a template and setting up an instance. So the code here does only the minimum necessary for reading the values.

```
607 \cs_new_protected:Npn _template_parse_values:nnn #1#2#3
608 {
609 _template_recover_keytypes:nn {#1} {#2}
610 \keyval_parse:NNn
611 _template_parse_values_elt:n _template_parse_values_elt:nn {#3}
612 }
```

(End of definition for `\_template_parse_values:nnn`.)

`\_template_parse_values_elt:n` Every key needs a value, so this is just an error routine.

```
613 \cs_new_protected:Npn _template_parse_values_elt:n #1
614 {
615 \bool_set_true:N \l__template_error_bool
616 \msg_error:nnn { template } { key-no-value } {#1}
617 }
```

(End of definition for `\_template_parse_values_elt:n`.)

To store the value, find the keytype then call the saving function. These need the current key name saved as `\l__template_key_name_tl`.

```
618 \cs_new_protected:Npn _template_parse_values_elt:nn #1#2
619 {
620 \tl_set:Ne \l__template_key_name_tl
621 { \tl_trim_spaces:e { \tl_to_str:n {#1} } }
622 \prop_get:NVNTF \l__template_keytypes_prop \l__template_key_name_tl
623 \l__template_tmp_tl
624 { _template_parse_values_elt_aux:n {#2} }
625 { \msg_error:nnV { template } { unknown-key } \l__template_key_name_tl }
626 }
627 \cs_new_protected:Npn _template_parse_values_elt_aux:n #1
628 {
629 _template_split_keytype_arg:V \l__template_tmp_tl
630 \use:c { __template_store_value_ \l__template_keytype_tl :n } {#1}
631 }
```

(End of definition for `\_template_parse_values_elt:nn` and `\_template_parse_values_elt_aux:n`.)

`\_template_template_set_eq:nnn` To copy a template, each of the lists plus the code has to be copied across. To keep this independent of the list storage system, it is all done with two-part shuffles.

```
632 \cs_new_protected:Npn _template_template_set_eq:nnn #1#2#3
633 {
634 _template_recover_defaults:nn {#1} {#3}
635 _template_store_defaults:nn {#1} {#2}
636 _template_recover_keytypes:nn {#1} {#3}
637 _template_store_keytypes:nn {#1} {#2}
638 _template_recover_vars:nn {#1} {#3}
639 _template_store_vars:nn {#1} {#2}
640 \cs_if_exist:cT { \c__template_code_root_tl #1 / #2 }
641 { \msg_info:nnnn { template } { declare-template-code } {#1} {#2} }
642 \cs_gset_eq:cc { \c__template_code_root_tl #1 / #2 }
643 { \c__template_code_root_tl #1 / #3 }
644 }
```

(End of definition for `\_template_template_set_eq:nnn`.)

## 12.8 Creating instances of templates

Making an instance has two distinct parts. First, the keys given are parsed to transfer the values into the structured data format used internally. This allows the default and given values to be combined with no repetition. In the second step, the structured data is converted to pre-defined variable assignments, and these are stored in the function for the instance.

```
645 \cs_new_protected:Npn _template_declare_instance:nnnn #1#2#3#4
646 {
647 _template_execute_if_code_exist:nnT {#1} {#2}
648 {
649 _template_recover_defaults:nn {#1} {#2}
650 _template_recover_vars:nn {#1} {#2}
651 _template_declare_instance_aux:nnnn {#1} {#2} {#3} {#4}
652 }
653 }
654 \cs_new_protected:Npn _template_declare_instance_aux:nnnn #1#2#3#4
655 {
656 \bool_set_false:N \l__template_error_bool
657 _template_parse_values:nnn {#1} {#2} {#4}
658 \bool_if:NF \l__template_error_bool
659 {
660 \prop_put:Nnn \l__template_values_prop { from-template } {#2}
661 _template_store_values:nn {#1} {#3}
662 _template_convert_to_assignments:
663 \cs_if_exist:cT { \c__template_instances_root_tl #1 / #3 }
664 { \msg_info:nnnn { template } { declare-instance } {#3} {#1} }
665 \cs_set_protected:cpe { \c__template_instances_root_tl #1 / #3 }
666 {
667 \exp_not:N _template_assignments_push:n
668 { \exp_not:V \l__template_assignments_tl }
669 \exp_not:c { \c__template_code_root_tl #1 / #2 }
670 }
671 }
672 }
```

(End of definition for `\_template_declare_instance:nnnn` and  
`\_template_declare_instance_aux:nnnn`.)

`\_template_instance_set_eq:nnn` Copy–paste an instance.

```
673 \cs_new_protected:Npn _template_instance_set_eq:nnn #1#2#3
674 {
675 _template_if_instance_exist:nnTF {#1} {#3}
676 {
677 _template_recover_values:nn {#1} {#3}
678 _template_store_values:nn {#1} {#2}
679 \cs_if_exist:cT { \c__template_instances_root_tl #1 / #2 }
680 { \msg_info:nnnn { template } { declare-instance } {#2} {#1} }
681 \cs_set_eq:cc { \c__template_instances_root_tl #1 / #2 }
682 { \c__template_instances_root_tl #1 / #3 }
683 }
684 { \msg_error:nnnn { template } { unknown-instance } {#1} {#3} }
```

```
685 }
```

(End of definition for `\_\_template\_instance\_set\_eq:nnn.`)

Editing an instance is almost identical to declaring one. The only variation is the source of the values to use. When editing, they are recovered from the previous instance run.

```
686 \cs_new_protected:Npn __template_edit_instance:nnn #1#2#3
687 {
688 __template_if_instance_exist:nnTF {#1} {#2}
689 {
690 __template_recover_values:nn {#1} {#2}
691 \prop_get:NnN \l__template_values_prop { from~template }
692 \l__template_tmp_tl
693 __template_edit_instance_aux:nVnn
694 {#1} \l__template_tmp_tl {#2} {#3}
695 }
696 { \msg_error:nnnn { template } { unknown-instance } {#1} {#2} }
697 }
698 \cs_new_protected:Npn __template_edit_instance_aux:nnnn #1#2#3#4
699 {
700 __template_recover_vars:nn {#1} {#2}
701 __template_declare_instance_aux:nnnn {#1} {#2} {#3} {#4}
702 }
703 \cs_generate_variant:Nn __template_edit_instance_aux:nnnn { nV }
```

(End of definition for `\_\_template_edit_instance:nnn` and `\_\_template_edit_instance_aux:nnnn.`)

The idea on converting to a set of assignments is to loop over each key, so that the loop order follows the declaration order of the keys. This is done using a sequence as property lists are not “ordered”.

```
704 \cs_new_protected:Npn __template_convert_to_assignments:
705 {
706 \tl_clear:N \l__template_assignments_tl
707 \seq_map_function:NN \l__template_key_order_seq
708 __template_convert_to_assignments_aux:n
709 }
710 \cs_new_protected:Npn __template_convert_to_assignments_aux:n #1
711 {
712 \prop_get:NnN \l__template_keytypes_prop {#1} \l__template_tmp_tl
713 __template_convert_to_assignments_aux:nV {#1} \l__template_tmp_tl
714 }
```

The second auxiliary function actually does the work. The arguments here are the key name (#1) and the keytype (#2). From those, the value to assign and the name of the appropriate variable are recovered. A bit of work is then needed to sort out keytypes with arguments (for example instances), and to look for global assignments. Once that is done, a hand-off can be made to the handler for the relevant keytype.

```
715 \cs_new_protected:Npn __template_convert_to_assignments_aux:nn #1#2
716 {
717 \prop_get:NnNT \l__template_values_prop {#1} \l__template_value_tl
718 {
719 \prop_get:NnNTF \l__template_vars_prop {#1} \l__template_var_tl
720 {
721 __template_split_keytype_arg:n {#2}
```

```

722 \str_if_eq:VnF \l__template_keytype_tl { choice }
723 {
724 \str_if_eq:VnF \l__template_keytype_tl { code }
725 { __template_find_global: }
726 }
727 \tl_set:Nn \l__template_key_name_tl {#1}
728 \cs_if_exist_use:cF { __template_assign_ \l__template_keytype_tl : }
729 { __template_assign_variable: }
730 }
731 { \msg_error:nnn { template } { unknown-attribute } {#1} }
732 }
733 }
734 \cs_generate_variant:Nn __template_convert_to_assignments_aux:nn { nV }

(End of definition for __template_convert_to_assignments:,
 __template_convert_to_assignments_aux:n, and __template_convert_to_assignments_aux:nn.)

```

\\_\_template\_find\_global: Global assignments should have the phrase `global` at the front. This is pretty easy to find: no other error checking, though.

```

735 \cs_new_protected:Npn __template_find_global:
736 {
737 \bool_set_false:N \l__template_global_bool
738 \tl_if_in:onT \l__template_var_tl { global }
739 {
740 \exp_after:wN __template_find_global_aux:w \l__template_var_tl \s__template_stop
741 }
742 }
743 \cs_new_protected:Npn __template_find_global_aux:w #1 global #2 \s__template_stop
744 {
745 \tl_set:Nn \l__template_var_tl {#2}
746 \bool_set_true:N \l__template_global_bool
747 }

```

(End of definition for \\_\_template\_find\_global: and \\_\_template\_find\_global\_aux:w.)

## 12.9 Using templates directly

Directly use a template with a particular parameter setting. This is also picked up if used in a nested fashion inside a parameter list. The idea is essentially the same as creating an instance, just with no saving of the result.

```

748 \cs_new_protected:Npn __template_use_template:nnn #1#2#3
749 {
750 __template_execute_if_code_exist:nnT {#1} {#2}
751 {
752 __template_recover_defaults:nn {#1} {#2}
753 __template_recover_vars:nn {#1} {#2}
754 __template_parse_values:nnn {#1} {#2} {#3}
755 __template_convert_to_assignments:
756 \use:c { \c__template_code_root_tl #1 / #2 }
757 }
758 }

```

(End of definition for \\_\_template\_use\_template:nnn.)

## 12.10 Assigning values to variables

\\_\\_template\\_assign\\_boolean:  
\\_\\_template\\_assign\\_boolean\\_aux:n

Setting a Boolean value is slightly different to everything else as the value can be used to work out which `set` function to call. As long as there is no need to recover things from another variable, everything is pretty easy. If there is, then we need to allow for the fact that the recovered value here will *not* be expandable, so needs to be converted to something that is.

```

759 \cs_new_protected:Npn __template_assign_boolean:
760 {
761 \bool_if:NTF \l__template_global_bool
762 { __template_assign_boolean_aux:n { bool_gset } }
763 { __template_assign_boolean_aux:n { bool_set } }
764 }
765 \cs_new_protected:Npn __template_assign_boolean_aux:n #
766 {
767 __template_if_key_value:VTF \l__template_value_tl
768 {
769 __template_key_to_value:
770 \tl_put_right:Ne \l__template_assignments_tl
771 {
772 \exp_not:c { #1 _eq:NN }
773 \exp_not:V \l__template_var_tl
774 \exp_not:V \l__template_value_tl
775 }
776 }
777 {
778 \tl_put_right:Ne \l__template_assignments_tl
779 {
780 \exp_not:c { #1 _ \l__template_value_tl :N }
781 \exp_not:V \l__template_var_tl
782 }
783 }
784 }
```

(End of definition for `\_\_template_assign_boolean:` and `\_\_template_assign_boolean_aux:n`.)

\\_\\_template\\_assign\\_choice:  
\\_\\_template\\_assign\\_choice\\_aux:nF  
\\_\\_template\\_assign\\_choice\\_aux:ef

The idea here is to find either the choice as-given or else the special `unknown` choice, and to copy the appropriate code across.

```

785 \cs_new_protected:Npn __template_assign_choice:
786 {
787 __template_assign_choice_aux:ef
788 { \l__template_key_name_tl \c_space_tl \l__template_value_tl }
789 {
790 __template_assign_choice_aux:ef
791 { \l__template_key_name_tl \c_space_tl unknown }
792 {
793 \prop_get:NVN \l__template_keytypes_prop \l__template_key_name_tl
794 \l__template_tmp_tl
795 __template_split_keytype_arg:V \l__template_tmp_tl
796 \msg_error:nnVV { template } { unknown-choice }
797 \l__template_key_name_tl
798 \l__template_value_tl
799 }
800 }
```

```

801 }
802 \cs_new_protected:Npn __template_assign_choice_aux:nF #1
803 {
804 \prop_get:NnNTF \l__template_vars_prop {#1} \l__template_tmp_t1
805 { \tl_put_right:NV \l__template_assignments_t1 \l__template_tmp_t1 }
806 }
807 \cs_generate_variant:Nn __template_assign_choice_aux:nF { e }

(End of definition for __template_assign_choice: and __template_assign_choice_aux:nF.)

```

\\_\_template\_assign\_function:  
\\_\_template\_assign\_function\_aux:N

```

808 \cs_new_protected:Npn __template_assign_function:
809 {
810 \bool_if:NTF \l__template_global_bool
811 { __template_assign_function_aux:N \cs_gset:Npn }
812 { __template_assign_function_aux:N \cs_set:Npn }
813 }
814 \cs_new_protected:Npn __template_assign_function_aux:N #1
815 {
816 \tl_put_right:Ne \l__template_assignments_t1
817 {
818 \cs_generate_from_arg_count:Nnnn
819 \exp_not:V \l__template_var_t1
820 \exp_not:N #1
821 { \exp_not:V \l__template_keytype_arg_t1 }
822 { \exp_not:V \l__template_value_t1 }
823 }
824 }

```

(End of definition for \\_\_template\_assign\_function: and \\_\_template\_assign\_function\_aux:N.)

\\_\_template\_assign\_instance:  
\\_\_template\_assign\_instance\_aux:N

```

825 \cs_new_protected:Npn __template_assign_instance:
826 {
827 \bool_if:NTF \l__template_global_bool
828 { __template_assign_instance_aux:N \cs_gset_protected:Npn }
829 { __template_assign_instance_aux:N \cs_set_protected:Npn }
830 }
831 \cs_new_protected:Npn __template_assign_instance_aux:N #1
832 {
833 \tl_put_right:Ne \l__template_assignments_t1
834 {
835 \exp_not:N #1 \exp_not:V \l__template_var_t1
836 {
837 __template_use_instance:nn
838 { \exp_not:V \l__template_keytype_arg_t1 }
839 { \exp_not:V \l__template_value_t1 }
840 }
841 }
842 }

```

(End of definition for \\_\_template\_assign\_instance: and \\_\_template\_assign\_instance\_aux:N.)

\\_\\_template\\_assign\\_variable:  
\\_\\_template\\_assign\\_variable:n  
A general-purpose function for all of the other assignments. As long as the value is not coming from another variable, the stored value is simply transferred for output. We use V-type expansion for the \KeyValue case: for token lists this is essential, whilst for register-based variables, it does no harm and avoids needing a low-level test.

```

843 \cs_new_protected:Npn __template_assign_variable:
844 {
845 \exp_args:Ne __template_assign_variable:n
846 {
847 __template_map_var_type:
848 -
849 \bool_if:NT \l__template_global_bool { g }
850 set:N
851 }
852 }
853 \cs_new_protected:Npn __template_assign_variable:n #1
854 {
855 __template_if_key_value:VTF \l__template_value_tl
856 {
857 __template_key_to_value:
858 \tl_put_right:Ne \l__template_assignments_tl
859 {
860 \exp_not:c { #1 V } \exp_not:V \l__template_var_tl
861 \exp_not:V \l__template_value_tl
862 }
863 }
864 {
865 \tl_put_right:Ne \l__template_assignments_tl
866 {
867 \exp_not:c { #1 n } \exp_not:V \l__template_var_tl
868 { \exp_not:V \l__template_value_tl }
869 }
870 }
871 }
```

(End of definition for \\_\\_template\\_assign\\_variable: and \\_\\_template\\_assign\\_variable:n.)

\\_\\_template\\_key\\_to\\_value:  
\\_\\_template\\_key\\_to\\_value\\_auxi:  
\\_\\_template\\_key\\_to\\_value\\_auxii:  
The idea here is to recover the attribute value of another key. To do that, the marker is removed and a look up takes place. If this is successful, then the name of the variable of the attribute is returned. This assumes that the value will be used in context where it will be converted to a value, for example when setting a number. There is also a need to check in case the copied value happens to be global.

```

872 \cs_new_protected:Npn __template_key_to_value:
873 { \exp_after:wN __template_key_to_value_auxi:w \l__template_value_tl }
874 \cs_new_protected:Npn __template_key_to_value_auxi:w \KeyValue #1
875 {
876 \tl_set:Ne \l__template_tmp_tl { \tl_trim_spaces:e { \tl_to_str:n {#1} } }
877 \prop_get:NVNTF \l__template_vars_prop \l__template_tmp_tl
878 \l__template_value_tl
879 {
880 \exp_after:wN __template_key_to_value_auxii:w \l__template_value_tl
881 \s__template_mark global \q__template_nil \s__template_stop
882 }
883 { \msg_error:nnV { template } { unknown-attribute } \l__template_tmp_tl }
```

```

884 }
885 \cs_new_protected:Npn __template_key_to_value_auxii:w #1 global #2#3 \s__template_stop
886 {
887 __template_quark_if_nil:NF #2
888 { \tl_set:Nn \l__template_value_tl {#2} }
889 }

```

(End of definition for `\__template_key_to_value:`, `\__template_key_to_value_auxi:w`, and  
`\__template_key_to_value_auxii:w`.)

## 12.11 Using instances

Using an instance is just a question of finding the appropriate function. If nothing is found, an error is raised. One complication is that if the first token of argument #2 is `\UseTemplate` then that is also valid. There is an error-test to make sure that the types agree, and if so the template is used directly.

```

890 \cs_new_protected:Npn __template_use_instance:nn #1#2
891 {
892 __template_if_use_template:nTF {#2}
893 { __template_use_instance_aux:nNnn {#1} #2 }
894 { __template_use_instance_aux:nn {#1} {#2} }
895 }
896 \cs_new_protected:Npn __template_use_instance_aux:nNnnn #1#2#3#4#5
897 {
898 \str_if_eq:nnTF {#1} {#3}
899 { __template_use_template:nnn {#3} {#4} {#5} }
900 { \msg_error:nNNNN { template } { type-mismatch } {#1} {#3} }
901 }
902 \cs_new_protected:Npn __template_use_instance_aux:nn #1#2
903 {
904 __template_if_instance_exist:nnTF {#1} {#2}
905 { \use:c { \c__template_instances_root_tl #1 / #2 } }
906 { \msg_error:nNNNN { template } { unknown-instance } {#1} {#2} }
907 }

```

(End of definition for `\__template_use_instance:nn`, `\__template_use_instance_aux:nNnnn`, and  
`\__template_use_instance_aux:nn`.)

## 12.12 Assignment manipulation

A few functions to transfer assignments about, as this is needed by `\AssignTemplateKeys`.

`\__template_assignments_pop`: To actually use the assignments.

```

908 \cs_new:Npn __template_assignments_pop: { \l__template_assignments_tl }

```

(End of definition for `\__template_assignments_pop`.)

`\__template_assignments_push`: Here, the assignments are stored for later use.

```

909 \cs_new_protected:Npn __template_assignments_push:n #1
910 { \tl_set:Nn \l__template_assignments_tl {#1} }

```

(End of definition for `\__template_assignments_push`.)

### 12.13 Showing templates and instances

\\_\\_template\\_show\\_code:nn

Showing the code for a template is just a translation of \cs\\_show:c.

```
911 \cs_new_protected:Npn __template_show_code:nn #1#2
912 { \cs_show:c { \c__template_code_root_tl #1 / #2 } }
```

(End of definition for \\_\\_template\\_show\\_code:nn.)

A modified version of the property-list printing code, such that the output refers to templates and instances rather than to the underlying structures.

```
913 \cs_new_protected:Npn __template_show_defaults:nn #1#2
914 {
915 __template_if_keys_exist:nnT {#1} {#2}
916 {
917 __template_recover_defaults:nn {#1} {#2}
918 __template_show:Nnnn \l__template_values_prop
919 {#1} {#2} { default-values }
920 }
921 }
922 \cs_new_protected:Npn __template_show_keytypes:nn #1#2
923 {
924 __template_if_keys_exist:nnT {#1} {#2}
925 {
926 __template_recover_keytypes:nn {#1} {#2}
927 __template_show:Nnnn \l__template_keytypes_prop
928 {#1} {#2} { interface }
929 }
930 }
931 \cs_new_protected:Npn __template_show_vars:nn #1#2
932 {
933 __template_execute_if_code_exist:nnT {#1} {#2}
934 {
935 __template_recover_vars:nn {#1} {#2}
936 __template_show:Nnnn \l__template_vars_prop
937 {#1} {#2} { variable-mapping }
938 }
939 }
940 \cs_new_protected:Npn __template_show:Nnnn #1#2#3#4
941 {
942 \msg_show:nneeee { template } { show-attribute }
943 { \t1_to_str:n {#2} }
944 { \t1_to_str:n {#3} }
945 { \t1_to_str:n {#4} }
946 { \prop_map_function:NN #1 \msg_show_item_unbraced:nn }
947 }
```

(End of definition for \\_\\_template\\_show\\_defaults:nn and others.)

\\_\\_template\\_show\\_values:nn

Instance values are a little more complex, as is the template to consider.

```
948 \cs_new_protected:Npn __template_show_values:nn #1#2
949 {
950 __template_if_instance_exist:nnT {#1} {#2}
951 {
952 __template_recover_values:nn {#1} {#2}
```

```

953 \msg_show:nneee { template } { show-values }
954 { \tl_to_str:n {#1} }
955 { \tl_to_str:n {#2} }
956 {
957 \prop_map_function:NN \l__template_values_prop
958 \msg_show_item_unbraced:nn
959 }
960 }
961 }
```

(End of definition for `\_template_show_values:nn`.)

## 12.14 Messages

The text for error messages: short and long text for all of them.

```

962 \msg_new:nnnn { template } { argument-number-mismatch }
963 { Template-type'#1'~takes~#2~argument(s). }
964 {
965 Templates~of~type~'#1'~require~#2~argument(s).\\
966 You~have~tried~to~make~a~template~for~'#1'~
967 with~#3~argument(s),~which~is~not~possible:~
968 the~number~of~arguments~must~agree.
969 }
970 \msg_new:nnnn { template } { bad-number-of-arguments }
971 { Bad~number~of~arguments~for~template-type'#1'. }
972 {
973 A~template~may~accept~between~0~and~9~arguments.\\
974 You~asked~to~use~#2~arguments:~this~is~not~supported.
975 }
976 \msg_new:nnnn { template } { bad-variable }
977 { Incorrect~variable~description'#1'. }
978 {
979 The~argument~'#1'~is~not~of~the~form~\\
980 ~~'<variable>'\\
981 ~or~\\
982 ~~'global~<variable>'.\\
983 It~must~be~given~in~one~of~these~formats~to~be~used~in~a~template.
984 }
985 \msg_new:nnnn { template } { choice-not-implemented }
986 { The~choice~'#1'~has~no~implementation. }
987 {
988 Each~choice~listed~in~the~interface~for~a~template~must~
989 have~an~implementation.
990 }
991 \msg_new:nnnn { template } { choice-no-code }
992 { The~choice~'#1'~requires~implementation~details. }
993 {
994 When~creating~template~code~using~\DeclareTemplateCode,~
995 each~choice~name~must~have~an~associated~implementation.\\
996 This~should~be~given~after~a~'='~sign:~LaTeX~did~not~find~one.
997 }
998 \msg_new:nnnn { template } { choice-requires-code }
999 { The~choice~'#2'~for~key~'#1'~requires~an~implementation. }
1000 }
```

```

1001 You~should~have~put:\\
1002 \\ \ #1-:~choice{-~#2 = <code> ~} \\
1003 but~LaTeX~did~not~find~any~<code>.
1004 }
1005 \msg_new:nnnn { template } { duplicate-key-interface }
1006 { Key-#1-appears-twice-in-interface-definition-\msg_line_context:.. }
1007 {
1008 Each-key-can-only-have-one-interface-declared-in-a-template.\\
1009 LaTeX-found-two-interfaces-for-'#1'.
1010 }
1011 \msg_new:nnnn { template } { keytype-requires-argument }
1012 { The-key-type-'#1-requires-an-argument-\msg_line_context:.. }
1013 {
1014 You~should~have~put:\\
1015 \\ \ <key-name>~-#1~-{~<argument>~} \\
1016 but~LaTeX~did~not~find~an~<argument>.
1017 }
1018 \msg_new:nnnn { template } { invalid-keytype }
1019 { The-key-'#1-~is~missing~a~key-type-\msg_line_context:.. }
1020 {
1021 Each-key-in-a-template-requires-a-key-type,-given-in-the-form:\\
1022 \\ \ <key>~-~<key-type>\\
1023 LaTeX~could~not~find~a~<key-type>~in~your~input.
1024 }
1025 \msg_new:nnnn { template } { key-no-value }
1026 { The-key-'#1-~has~no~value-\msg_line_context:.. }
1027 {
1028 When~creating~an~instance~of~a~template~
1029 every~key~listed~must~include~a~value:\\
1030 \\ \ <key>~-~<value>
1031 }
1032 \msg_new:nnnn { template } { key-no-variable }
1033 { The-key-'#1-~requires~implementation-details-\msg_line_context:.. }
1034 {
1035 When~creating~template~code~using~\DeclareTemplateCode,~
1036 each~key~name~must~have~an~associated~implementation.\\
1037 This~should~be~given~after~a-='~sign:~LaTeX~did~not~find~one.
1038 }
1039 \msg_new:nnnn { template } { key-not-implemented }
1040 { Key-'#1-~has~no~implementation-\msg_line_context:.. }
1041 {
1042 The-definition-of-key-implementations-for-template-'#2'-
1043 of-template-type-'#3'-~does~not~include~any~details~for~key-'#1'.\\
1044 The-key~was~declared~in~the~interface~definition,~
1045 and~so~an~implementation~is~required.
1046 }
1047 \msg_new:nnnn { template } { missing-keytype }
1048 { The-key-'#1-~is~missing~a~key-type-\msg_line_context:.. }
1049 {
1050 Key-interface~definitions~should~be~of~the~form\\
1051 \\ \ #1-:~<key-type>\\
1052 but~LaTeX~could~not~find~a~<key-type>.
1053 }
1054 \msg_new:nnnn { template } { no-template-code }

```

```

1055 {
1056 The~template~'#2'~of~type~'#1'~is~unknown~
1057 or~has~no~implementation.
1058 }
1059 {
1060 There~is~no~code~available~for~the~template~name~given.\\
1061 This~should~be~given~using~\DeclareTemplateCode.
1062 }
1063 \msg_new:nnnn { template } { type-already-defined }
1064 { Template-type~'#1'~already-defined. }
1065 {
1066 You~have~used~\NewTemplateType~
1067 with~a~template~type~that~has~already~been~defined.
1068 }
1069 \msg_new:nnnn { template } { type-mismatch }
1070 { Template-types~'#1'~and~'#2'~do~not~agree. }
1071 {
1072 You~are~trying~to~use~a~template~directly~with~\UseInstance
1073 (or~a~similar~function),~but~the~template~types~do~not~match.
1074 }
1075 \msg_new:nnnn { template } { unknown-attribute }
1076 { The~template~attribute~'#1'~is~unknown. }
1077 {
1078 There~is~a~definition~in~the~current~template~reading\\
1079 \\ \\ \token_to_str:N \KeyValue {~#1~} \\
1080 but~there~is~no~key~called~'#1'.
1081 }
1082 \msg_new:nnnn { template } { unknown-choice }
1083 { The~choice~'#2'~was~not~declared~for~key~'#1'. }
1084 {
1085 The~key~'#1'~takes~a~fixed~list~of~choices~
1086 and~this~list~does~not~include~'#2'.
1087 }
1088 \msg_new:nnnn { template } { unknown-default-choice }
1089 { The~default~choice~'#2'~was~not~declared~for~key~'#1'. }
1090 {
1091 The~key~'#1'~takes~a~fixed~list~of~choices~
1092 and~this~list~does~not~include~'#2'.
1093 }
1094 \msg_new:nnnn { template } { unknown-instance }
1095 { The~instance~'#2'~of~type~'#1'~is~unknown. }
1096 {
1097 You~have~asked~to~use~an~instance~'#2',~
1098 but~this~has~not~been~created.
1099 }
1100 \msg_new:nnnn { template } { unknown-key }
1101 { Unknown~template~key~'#1'. }
1102 {
1103 The~key~'#1'~was~not~declared~in~the~interface~
1104 for~the~current~template.
1105 }
1106 \msg_new:nnnn { template } { unknown-keytype }
1107 { The~key~type~'#1'~is~unknown. }
1108 {

```

```

1109 Valid-key-types-are:\\
1110 --boolean;\\
1111 --choice;\\
1112 --commalist;\\
1113 --function;\\
1114 --instance;\\
1115 --integer;\\
1116 --length;\\
1117 --muskip;\\
1118 --real;\\
1119 --skip;\\
1120 --tokenlist.
1121 }
1122 \msg_new:nnn { template } { unknown-type }
1123 { The-template-type-'#1'~is~unknown. }
1124 {
1125 A-template-type-needs-to-be-defined-with-\NewTemplateType
1126 prior-to-using-it.
1127 }
1128 \msg_new:nnn { template } { unknown-template }
1129 { The-template-'#2'~of~type-'#1'~is~unknown. }
1130 {
1131 No-interface-has-been-declared-for-a-template-
1132 '#2'~of~template-type-'#1'.
1133 }

```

Information messages only have text: more text should not be needed.

```

1134 \msg_new:nnn { template } { declare-instance }
1135 { Declaring-instance~~'#1'~of~type~#2~\msg_line_context:.. }
1136 \msg_new:nnn { template } { declare-template-code }
1137 { Declaring-code-for-template~'#2'~of~template-type~'#1'~\msg_line_context:.. }
1138 \msg_new:nnn { template } { declare-template-interface }
1139 {
1140 Declaring-interface-for-template~'#2'~of~template-type~'#1'~
1141 \msg_line_context:..
1142 }
1143 \msg_new:nnn { template } { declare-type }
1144 { Declaring-template-type~'#1'~taking~#2~argument(s)~\msg_line_context:.. }
1145 \msg_new:nnn { template } { show-attribute }
1146 {
1147 The-template~'#2'~of~type~'#1'~has~
1148 \tl_if_empty:nTF {#4} { no-#3. } { #3 : #4 }
1149 }
1150 \msg_new:nnn { template } { show-values }
1151 {
1152 The-instance~'#2'~of~type~'#1'~has~
1153 \tl_if_empty:nTF {#3} { no-values. } { values: #3 }
1154 }

```

Also add template to the  $\text{\LaTeX}$  messages.

```

1155 \prop_gput:Nnn \g_msg_module_type_prop { template } { \LaTeX }

```

## 12.15 User functions

All simple translations.

```

\NewTemplateType
\DeclareTemplateInterface
\DeclareTemplateCode
\DeclareTemplateCopy
>EditTemplateDefaults
 \UseTemplate
 \DeclareInstance
\DeclareInstanceCopy
 \EditInstance
 \UseInstance

```

```

1156 \cs_new_protected:Npn \NewTemplateType #1#2
1157 { __template_define_type:nn {#1} {#2} }
1158 \cs_new_protected:Npn \DeclareTemplateInterface #1#2#3#4
1159 { __template_declare_template_keys:nnnn {#1} {#2} {#3} {#4} }
1160 \cs_new_protected:Npn \DeclareTemplateCode #1#2#3#4#5
1161 { __template_declare_template_code:nnnnn {#1} {#2} {#3} {#4} {#5} }
1162 \cs_new_protected:Npn \DeclareTemplateCopy #1#2#3
1163 { __template_template_set_eq:nnn {#1} {#2} {#3} }
1164 \cs_new_protected:Npn \EditTemplateDefaults #1#2#3
1165 { __template_edit_defaults:nnn {#1} {#2} {#3} }
1166 \cs_new_protected:Npn \UseTemplate #1#2#3
1167 { __template_use_template:nnm {#1} {#2} {#3} }
1168 \cs_new_protected:Npn \DeclareInstance #1#2#3#4
1169 { __template_declare_instance:nnnn {#1} {#3} {#2} {#4} }
1170 \cs_new_protected:Npn \DeclareInstanceCopy #1#2#3
1171 { __template_instance_set_eq:nnn {#1} {#2} {#3} }
1172 \cs_new_protected:Npn \EditInstance #1#2#3
1173 { __template_edit_instance:nnn {#1} {#2} {#3} }
1174 \cs_new_protected:Npn \UseInstance #1#2
1175 { __template_use_instance:nn {#1} {#2} }

```

(End of definition for `\NewTemplateType` and others. These functions are documented on page 343.)

The show functions are again just translation.

```

>ShowTemplateCode
>ShowTemplateDefaults
>ShowTemplateInterface
>ShowTemplateVariables
>ShowInstanceValues

```

```

1176 \cs_new_protected:Npn \ShowTemplateCode #1#2
1177 { __template_show_code:nn {#1} {#2} }
1178 \cs_new_protected:Npn \ShowTemplateDefaults #1#2
1179 { __template_show_defaults:nn {#1} {#2} }
1180 \cs_new_protected:Npn \ShowTemplateInterface #1#2
1181 { __template_show_keytypes:nn {#1} {#2} }
1182 \cs_new_protected:Npn \ShowTemplateVariables #1#2
1183 { __template_show_vars:nn {#1} {#2} }
1184 \cs_new_protected:Npn \ShowInstanceValues #1#2
1185 { __template_show_values:nn {#1} {#2} }

```

(End of definition for `\ShowTemplateCode` and others. These functions are documented on page 350.)

More direct translation.

```

>IfInstanceExistsT
>IfInstanceExistsF
>IfInstanceExistsTF

```

```

1186 \cs_new:Npn \IfInstanceExistsTF #1#2
1187 { __template_if_instance_exist:nnTF {#1} {#2} }
1188 \cs_new:Npn \IfInstanceExistsT #1#2
1189 { __template_if_instance_exist:nnT {#1} {#2} }
1190 \cs_new:Npn \IfInstanceExistsF #1#2
1191 { __template_if_instance_exist:nnF {#1} {#2} }

```

(End of definition for `\IfInstanceExistsT`, `\IfInstanceExistsF`, and `\IfInstanceExistsTF`. These functions are documented on page 348.)

`\KeyValue` Simply dump the argument when executed: this should not happen.

```

1192 \cs_new_protected:Npn \KeyValue #1 {#1}

```

(End of definition for `\KeyValue`. This function is documented on page 345.)

**\AssignTemplateKeys** A short call to use a token register by proxy.

```
1193 \cs_new_protected:Npn \AssignTemplateKeys { _template_assignments_pop: }
```

(End of definition for **\AssignTemplateKeys**. This function is documented on page 345.)

**\SetTemplateKeys** A friendly wrapper

```
1194 \cs_new_protected:Npn \SetTemplateKeys #1#2#3
 { \keys_set_known:nnN { template / #1 / #2 } {#3} \l_template_tmp_clist }
```

(End of definition for **\SetTemplateKeys**. This function is documented on page 349.)

```
1196 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}{lttemplates}%
1197 ⟨latexrelease⟩ {Prototype~document~commands}%
1198 ⟨latexrelease⟩
1199 ⟨latexrelease⟩\EndModuleRelease
1200 \ExplSyntaxOff
1201 ⟨/2ekernel | latexrelease⟩
```

We need to stop DocStrip treating `@@` in a special way at this point.

```
1202 ⟨@=⟩
```

# File 12

## ltalloc.dtx

### 1 Counters

This section deals with counter and other variable allocation.

1  {\*2ekernel}

The following are from plain T<sub>E</sub>X:

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

\@ne The number 1.

\m@ne The number -1.

\tw@ The number 2.

\sixt@@n The number 16.

\@m The number 1000.

\@MM The number 20000.

\@xxxii The constant 32.

2  \chardef\@xxxii=32

(End of definition for \@xxxii.)

\@Mi Constants 10001–10004.

\@Mii 3  \mathchardef\@Mi=10001

\@Miii 4  \mathchardef\@Mii=10002

\@Miv 5  \mathchardef\@Miii=10003

6  \mathchardef\@Miv=10004

(End of definition for \@Mi and others.)

\@tempcnta Scratch count registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

\@tempcntb 7  \newcount\@tempcnta

8  \newcount\@tempcntb

(End of definition for \@tempcnta and \@tempcntb.)

\if@tempswa General boolean switch used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

9  \newif\if@tempswa

(End of definition for \if@tempswa.)

\@tempdima Scratch dimen registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.

\@tempdimb 10  \newdimen\@tempdima

\@tempdimc 11  \newdimen\@tempdimb

12  \newdimen\@tempdimc

(End of definition for \@tempdima, \@tempdimb, and \@tempdimc.)

\@tempboxa Scratch box register used by L<sup>A</sup>T<sub>E</sub>X kernel commands.  
   <sup>13</sup> \newbox\@tempboxa  
   *(End of definition for \@tempboxa.)*

\@tempskipa Scratch skip registers used by L<sup>A</sup>T<sub>E</sub>X kernel commands.  
   \@tempskipb  
   <sup>14</sup> \newskip\@tempskipa  
   <sup>15</sup> \newskip\@tempskipb  
   *(End of definition for \@tempskipa and \@tempskipb.)*

\@temptokena Scratch token register used by L<sup>A</sup>T<sub>E</sub>X kernel commands.  
   <sup>16</sup> \newtoks\@temptokena  
   *(End of definition for \@temptokena.)*

\@flushglue Glue used for \right- & \leftskip = 0pt plus 1fil  
   <sup>17</sup> \newskip\@flushglue \@flushglue = 0pt plus 1fil  
   *(End of definition for \@flushglue.)*  
   <sup>18</sup> ⟨/2ekernel⟩

# File 13

## ltcntrl.dtx

### 1 Program control structure

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

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
1 {*2ekernel}
2 \message{control,}

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

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

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

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

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

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

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

```

 \cnextwhile = def(\@iwhilenum)
else \cnextwhile = def(\@whilenoop)
fi
\cnextwhile {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
\cnextwhile = def(\@iwhilesw)
else \cnextwhile = def(\@whileswnoop)
fi
\cnextwhile {SWITCH BODY} \fi
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

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

```

*(End of definition for \@whilenoop , \@whilenum , and \@iwhilenum.)*

```

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

```

*(End of definition for \@whiledim and \@iwhiledim.)*

```

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

```

(End of definition for \@whileswnoop, \@whilesw, and \@iwhilesw.)

Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):

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

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

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

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

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

End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.

\@nnil

<sup>13</sup> \def\@nnil{\@nil}

(End of definition for \@nnil.)

\@empty

<sup>14</sup> \def\@empty{}

(End of definition for \@empty.)

```

\@fornoop
15 \long\def\@fornoop#1\@@#2#3{%
(End of definition for \@fornoop.)}

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

\@forloop
20 \long\def\@forloop#1,#2,#3\@@#4#5{\def#4{#1}\ifx #4\@nnil \else
21 #5\def#4{#2}\ifx #4\@nnil \else#5\@iforloop #3\@@#4{#5}\fi\fi}
(End of definition for \@forloop.)

\@iforloop
22 \long\def\@iforloop#1,#2\@@#3#4{\def#3{#1}\ifx #3\@nnil
23 \expandafter\@fornoop \else
24 #4\relax\expandafter\@iforloop\fi#2\@@#3{#4}}
(End of definition for \@iforloop.)

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

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

\@removeelement Removes an element from a comma-separated list and puts it into a control sequence,
called as \@removeelement{\<element>}{\<list>}{\<cs>}. Due to the implementation
method the \<element> is not allowed to contain braces.
32 \def\@removeelement#1#2#3{%
33 \def\reserved@a##1,#1,##2\reserved@a{##1,##2\reserved@b}%
34 \def\reserved@b##1,\reserved@b##2\reserved@b{%
35 \ifx,\#1\empty\else##1\fi}%
36 \edef#3{%
37 \expandafter\reserved@b\reserved@a,#2,\reserved@b,#1,\reserved@a}}
(End of definition for \@removeelement.)
38 </2ekernel>

```

# File 14

## lterror.dtx

### 1 Error handling and tracing

This section defines L<sup>A</sup>T<sub>E</sub>X's error commands.

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

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

These days we don't support autoloading approach any longer, but this part bit is kept in case it is used in old documents.

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

#### 1.1 General commands

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

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

(End of definition for \MessageBreak.)

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

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

(End of definition for \GenericInfo.)

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

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

(End of definition for \GenericWarning.)

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

```

18 \bgroup
19 \lccode`@`\ %
20 \lccode`~-`\ %
21 \lccode`]=`\ %
22 \lccode`{`\ %
23 \lccode`T`\T%
24 \lccode`H`\H%
25 \catcode`\\=11\relax%
26 \lowercase{%
27 \egroup%

```

Unfortunately 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`.

```

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

```

First the ‘standard case’.

```

30 \DeclareRobustCommand{\GenericError}[4]{%
31 \begingroup%
32 \immediate\write\@unused{}%
33 \def\MessageBreak{^^J}%
34 \set@display@protect%
35 \edef%
36 % %<-----do not delete this space!----->%
37 \err@%
38 {{#4}}%
39 \errhelp%
40 % %<-----do not delete this space!----->%
41 \err@%
42 \let%
43 % %<-----do not delete this space!----->%
44 \err@%
45 \empty%
46 \def\MessageBreak{^^J#1}%
47 \def~{\errmessage{%
48 #2.^^J^^J%
49 #3^^J%
50 Type H <return> for immediate help%
51 % %<-----do not delete this space!----->%
52 \err@%

```

```

53 } } %
54 ~%
55 \endgroup}%
56 \else%
 Secondly the version for old TEX's.
57 \DeclareRobustCommand{\GenericError}[4]{%
58 \begingroup%
59 \immediate\write\@unused{ }%
60 \def\MessageBreak{^ }%
61 \set@display@protect%
62 \edef%
63 % %<-----do not delete this space!----->%
64 \err@ %
65 {{#4}}%
66 \errhelp%
67 % %<-----do not delete this space!----->%
68 \err@ %
69 \let%
70 % %<-----do not delete this space!----->%
71 \err@ %
72 \errmessage%
73 \def\MessageBreak{^ J#1}%
74 \def~{\typeout{! } %
75 #2.^ J^ J%
76 #3^ J%
77 Type H <return> for immediate help.)%
78 % %<-----do not delete this space!----->%
79 \err@ %
80 {}}%
81 ~%
82 \endgroup}%
83 \fi}%

```

(End of definition for \GenericError.)

|                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \PackageError<br>\PackageWarning<br>\PackageWarningNoLine<br>\PackageInfo<br>\ClassError<br>\ClassWarning<br>\ClassWarningNoLine<br>\ClassInfo | These commands are intended for use by package and class writers, to give information to authors. The syntax is: <pre> \PackageError{&lt;package&gt;}{&lt;error&gt;}{&lt;help&gt;} \PackageWarning{&lt;package&gt;}{&lt;warning&gt;} \PackageWarningNoLine{&lt;package&gt;}{&lt;warning&gt;} \PackageInfo{&lt;package&gt;}{&lt;info&gt;} </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

and similarly for classes. The **Error** commands print the **<error>** message, and present the interactive prompt; if the author types **h**, then the **<help>** information is displayed. The **Warning** commands produce a warning but do not present the interactive prompt. The **WarningNoLine** commands do the same, but don't print the input line number. The **Info** commands write the message to the **log** file. Within the messages, the command **\MessageBreak** can be used to break a line, **\protect** can be used to protect command names, and **\space** is a space, for example:

```
\newcommand{\foo}{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.

84 \gdef\PackageError#1#2#3{%
85 \GenericError{%
86 (#1)\@spaces\@spaces\@spaces\@spaces
87 }{%
88 Package #1 Error: #2%
89 }{%
90 See the #1 package documentation for explanation.%}
91 }{#3}%
92 }

93 \def\PackageWarning#1#2{%
94 \GenericWarning{%
95 (#1)\@spaces\@spaces\@spaces\@spaces
96 }{%
97 Package #1 Warning: #2%
98 }{%
99 }
100 \def\PackageWarningNoLine#1#2{%
101 \PackageWarning{#1}{#2\@gobble}%
102 }
103 \def\PackageInfo#1#2{%
104 \GenericInfo{%
105 (#1) \@spaces\@spaces\@spaces
106 }{%
107 Package #1 Info: #2%
108 }{%
109 }
110 \gdef\ClassError#1#2#3{%
111 \GenericError{%
112 (#1) \space\@spaces\@spaces\@spaces
113 }{%
114 Class #1 Error: #2%
115 }{%
116 See the #1 class documentation for explanation.%}
117 }{#3}%
118 }

119 \def\ClassWarning#1#2{%
120 \GenericWarning{%
121 (#1) \space\@spaces\@spaces\@spaces
122 }{%
123 Class #1 Warning: #2%
124 }{%
125 }
126 \def\ClassWarningNoLine#1#2{%
```

```

127 \ClassWarning{#1}{#2\@gobble}%
128 }
129 \def\ClassInfo#1#2{%
130 \GenericInfo{%
131 (#1) \space\space\@spaces\@spaces
132 }{%
133 Class #1 Info: #2%
134 }%
135 }

```

(End of definition for `\PackageError` and others.)

```

\ClassNote
\ClassNoteNoLine 136 </2ekernel>
\PackageNote 137 <*2ekernel | latexrelease>
\PackageNoteNoLine 138 <| latexrelease>\IncludeInRelease{2021/11/15}%
139 <| latexrelease> {\ClassNote}{Notes for classes/packages}%
\def\ClassNote#1#2{%
141 \GenericWarning{%
142 (#1) \space\space\@spaces\@spaces
143 }{%
144 Class #1 Info: #2%
145 }%
146}
147 \def\ClassNoteNoLine#1#2{\ClassNote{#1}{#2\@gobble}}
\def\PackageNote#1#2{%
148 \GenericWarning{%
149 (#1) \@spaces\@spaces\@spaces
150 }{%
151 Package #1 Info: #2%
152 }%
153}
154 \def\PackageNoteNoLine#1#2{\PackageNote{#1}{#2\@gobble}}
155 </2ekernel | latexrelease>
156 <| latexrelease>\EndIncludeInRelease
157

```

We don't roll back, because if this code is used by packages then most often they will not have rollback code implemented, so they would immediately break even if they otherwise would work fine.

```

158 <| latexrelease>\IncludeInRelease{0000/00/00}%
159 <| latexrelease> {\ClassNote}{Notes for classes/packages}%
160 <| latexrelease>
161 <| latexrelease>\EndIncludeInRelease
162 <*2ekernel>

```

(End of definition for `\ClassNote` and others.)

|                                      |                                                                             |
|--------------------------------------|-----------------------------------------------------------------------------|
| <code>\@latex@error</code>           | Errors and other info, for use in the L <sup>A</sup> T <sub>E</sub> X core. |
| <code>\@latex@warning</code>         |                                                                             |
| <code>\@latex@warning@no@line</code> |                                                                             |
| <code>\@latex@info</code>            |                                                                             |
| <code>\@latex@info@no@line</code>    |                                                                             |

```

163 \gdef\@latex@error#1#2{%
164 \GenericError{%
165 \space\space\space\@spaces\@spaces\@spaces
166 }{%
167 LaTeX Error: #1%
168 }{%

```

```

169 See the LaTeX manual or LaTeX Companion for explanation.%

170 }{#2}%
171 }
172 \def\@latex@warning#1{%
173 \GenericWarning{%
174 \space\space\space\@spaces\@spaces\@spaces
175 }{%
176 LaTeX Warning: #1%
177 }%
178 }
179 \def\@latex@warning@no@line#1{%
180 \@latex@warning{#1\@gobble}}
181 \def\@latex@info#1{%
182 \GenericInfo{%
183 \@spaces\@spaces\@spaces
184 }{%
185 LaTeX Info: #1%
186 }%
187 }
188 \def\@latex@info@no@line#1{%
189 \@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 of definition for \@latex@error and others.)

\@latex@note These are “info” messages that display on the terminal not just in the transcript.

```

190 (/2ekernel)
191 (*2ekernel | latexrelease)
192 (<latexrelease>) \IncludeInRelease{2021/11/15}%
193 (<latexrelease>) {\@latex@note}{Display notes}%
194 \def\@latex@note#1{%
195 \GenericWarning{%
196 \@spaces\@spaces\@spaces
197 }{%
198 LaTeX Info: #1%
199 }%
200 }

```

```

201 \def\@latex@note@no@line#1{%
202 \@latex@note{#1\@gobble}}

```

We don't make them undefined but rather point to `\@latex@info` because that's what they replace. This way we can change `\@latex@info` elsewhere without the need to further rollback sections.

```

203 </2ekernel | latexrelease>
204 <latexrelease>\EndIncludeInRelease
205 <latexrelease>\IncludeInRelease{0000/00/00}%
206 <latexrelease> {\@latex@note}{Display notes}%
207 <latexrelease>
208 <latexrelease>\let\@latex@note\@latex@info
209 <latexrelease>\let\@latex@note@no@line\@latex@info@no@line
210 <latexrelease>\EndIncludeInRelease
211 <*2ekernel>

```

(End of definition for `\@latex@note` and `\@latex@note@no@line`.)

`\c@errorcontextlines` `\errorcontextlines` as a L<sup>A</sup>T<sub>E</sub>X counter, so that it may be manipulated with `\setcounter` (once it is defined :-)

```

212 \let\c@errorcontextlines\errorcontextlines
213 \c@errorcontextlines=-1

```

(End of definition for `\c@errorcontextlines`.)

`\on@line` The message ‘on input line *n*’.

```

214 \def\on@line{ on input line \the\inputlineno}

```

(End of definition for `\on@line`.)

`\@warning` `\@@warning` Older L<sup>A</sup>T<sub>E</sub>X messages. For the moment, these `\let` to the new message commands. They may be changed later, once only obsolete packages and classes contain them.

```

215 \let\@warning\@latex@warning
216 \let\@@warning\@latex@warning@no@line
217 \global\let\@latexerr\@latex@error

```

(End of definition for `\@warning`, `\@@warning`, and `\@latexerr`.)

`\@spaces` Four spaces.

```

218 \def\@spaces{\space\space\space\space}

```

(End of definition for `\@spaces`.)

## 1.2 Specific errors

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

```

219 \gdef\@ehaf{%
220 Your command was ignored.\MessageBreak
221 Type \space I <command> <return> \space to replace it %
222 with another command,\MessageBreak
223 or \space <return> \space to continue without it.}
224 \gdef\@ehbf{%
225 You've lost some text. \space \@ehc}
226 \gdef\@ehcf{%
227 Try typing \space <return> %

```

```

228 \space to proceed.\MessageBreak
229 If that doesn't work, type \space X <return> \space to quit.}
230 \gdef\@ehd{%
231 You're in trouble here. \space\@ehc}

```

(End of definition for \@eha and others.)

- \@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....
- ```

232  \gdef\@notdefinable{%
233    \@latex@error{%
234      Command \backslashreserved@a\space
235      already defined.\MessageBreak
236      Or name \backslashqend... illegal,
237      see p.192 of the manual}\@eha}

```

(End of definition for \@notdefinable.)

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

```

238  \gdef\@nolnerr{%
239    \@latex@error{There's no line here to end}\@eha}

```

(End of definition for \@nolnerr.)

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

\@nocnterr Obsolete error message generated in L^AT_EX2.09 by \setcounter, \addtocounter or \newcounter for undefined counter. DO NOT use for L^AT_EX2 _{ε} it MIGHT vanish! Use \@nocounterr{<cnt>} instead.

```

240  \gdef\@nocnterr#1{%
241    \@latex@error{No counter '#1' defined}\@eha}
242  \gdef\@nocnterr{\@nocounterr?}

```

(End of definition for \@nocounterr and \@nocnterr.)

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

```

243  \gdef\@ctrerr{%
244    \@latex@error{Counter too large}\@ehb}

```

(End of definition for \@ctrerr.)

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

```

245  \gdef\@nodocument{%
246    \@latex@error{Missing \protect\begin{document}}}\@ehd}

```

(End of definition for \@nodocument.)

\@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.

The environment name has to literally match, i.e., what is stored in \@currenvir (after one expansion) must match what is passed to \end (without expansion). If not we complain. Not the absolute best solution but at least it avoids getting \begin{foo} ended by \end{foo} which was possible in the past.

```
247 \gdef\@badend#1{%
248   \@latex@error{\protect\begin
249     {\detokenize\expandafter{\@currenvir}}\@currenvline
250     \space ended by \protect\end{\detokenize{#1}}}\@eha}
```

(End of definition for \@badend.)

\@badmath Called by \[, \], \{ or \} when used in wrong mode.

```
251 \gdef\@badmath{%
252   \@latex@error{Bad math environment delimiter}\@eha}
```

(End of definition for \@badmath.)

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

```
253 \gdef\@toodeep{%
254   \@latex@error{Too deeply nested}\@ehd}
```

(End of definition for \@toodeep.)

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

```
255 \gdef\@badpoptabs{%
256   \@latex@error{\protect\pushtabs\space and \protect\poptabs
257   \space don't match}\@ehd}
```

(End of definition for \@badpoptabs.)

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

```
258 \gdef\@badtab{%
259   \@latex@error{Undefined tab position}\@ehd}
```

(End of definition for \@badtab.)

\@preamerr This error is special: it appears in places where we normally have to \protect expansions. However, to prevent a protection of the error message itself (which would result in the message getting printed not issued on the terminal) we need to locally reset \protect to \relax.

```
260 \gdef\@preamerr#1{%
261   \begingroup
262   \let\protect\relax
263   \@latex@error{\ifcase #1 Illegal character\or
264     Missing @-exp\or Missing p-arg\fi\space
265     in array arg}\@ehd
266   \endgroup}
```

(End of definition for \@preamerr.)

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

```
267 \gdef\@badlinearg{%
268   \@latex@error{%
269     Bad \protect\line\space or \protect\vector
270     \space argument}\@ehb}
```

(End of definition for `\@badlinearg`.)

\@LRmoderr A command is not allowed in restricted horizontal mode, i.e., in LR-mode in L^AT_EX terminology.

```
271 \gdef\@LRmoderr{%
272   \@latex@error{Not allowed in LR mode}\@ehb}
```

(End of definition for `\@LRmoderr`.)

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

```
273 \gdef\@parmoderr{%
274   \@latex@error{Not in outer par mode}\@ehb}
```

(End of definition for `\@parmoderr`.)

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

```
275 \gdef\@fltovf{%
276   \@latex@error{Too many unprocessed floats}\@ehb}
```

(End of definition for `\@fltovf`.)

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

```
277 \gdef\@latexbug{%
278   \@latex@error{This may be a LaTeX bug}{Call for help}}
```

(End of definition for `\@latexbug`.)

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

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

(End of definition for `\@badcrerr`.)

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

```
280 \gdef\@noitemerr{%
281   \@latex@error{Something's wrong--perhaps a missing %
282     \protect\item}\@ehc}
```

(End of definition for `\@noitemerr`.)

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

```
283 \gdef\@notprerr{%
284   \@latex@error{Can be used only in preamble}\@eha}
```

(End of definition for `\@notprerr`.)

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

```
285 \gdef\@inmatherr#1{%
286   \relax
287   \ifmmode
288     \@latex@error{Command \protect#1 invalid in math mode}\@ehc
289   \fi}
```

(End of definition for `\@inmatherr`.)

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

```
290 %\def\@invalidchar{\@latex@error{Invalid character in input}\@ehc}
```

(End of definition for `\@invalidchar`.)

As well as the above error commands some error messages are directly coded to save space. The messages already present in L^AT_EX2.09 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 `\conditionally@traceon`.

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

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

\conditionally@traceon These are only dummy definitions. For details see the `trace` package.

```
291 \let\conditionally@traceon\@empty
292 \let\conditionally@traceoff\@empty
```

(End of definition for `\conditionally@traceon` and `\conditionally@traceoff`.)

```
293 </2ekernel>
```

File 15

ltpar.dtx

1 Paragraphs

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

This file here describes the interfaces that have been in the kernel forever, used to implement the scenarios described below. They remain valid but are now augmented in the next file (`ltpara.dtx`) to add hooks to paragraphs. At some point we will consolidate the two files further.

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.

1.1 Implementation

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

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

- `@nobreak`
- `@minipage`

they should do the setting if necessary.

```

1  {*2ekernel}
2  \message{par ,}

```

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

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

```

4  \def\@par{\let\par\@@par\par}

```

(End of definition for `\@setpar` and `\@par`.)

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

```

5  \def\@restorepar{\def\par{\@par}}
6  {/2ekernel}

```

(End of definition for `\@restorepar`.)

File 16

ltpara.dtx

Abstract

This code defines four special kernel hooks to support paragraph tagging as well as four public hooks which can be occasionally useful.

1 Introduction

The building of paragraphs in the T_EX engine(s) has a number of peculiarities that makes it on one hand fairly flexible but on the other hand somewhat awkward to control or reliably to extend. Thus to better understand the code below we start with a brief introduction of the mechanism; for more details refer to the T_EXbook [?, chap. 14] (for the full truth you may even have to study the program code).

1.1 The default processing done by the engine

T_EX automatically starts building a paragraph when it is currently in vertical mode and encounters anything that can only live in horizontal mode. Most often this is a character, but there are also many commands that can be used only in horizontal mode. If any of them is encountered, T_EX will immediately back up (i.e., the character or command is read later again), adds a `\parskip` glue to the current vertical list unless the list is empty, switches to horizontal mode, starts its special “start of paragraph processing” and only then rereads the character or command that caused the mode change.²³

This “start of paragraph processing” first adds an empty box at the start of the horizontal list of width `\parindent` (which represents the paragraph indentation) unless the paragraph was started with `\noindent` in which case no such box is added²⁴. It then reads and processes all tokens stored in the special engine token register `\everypar`. After that it reads and processes whatever has caused the paragraph to start.

Thus out of the box, T_EX offers the possibility to put some special code into `\everypar` to gain control at (more or less) the start of the paragraph. For example, in La_Te_X and a number of packages, special code like the following is sometimes used:

```
\everypar{{\setbox\z@\lastbox}\everypar{} ...}
```

This removes the paragraph indentation box again (that was already placed by T_EX), then resets `\everypar` so that it doesn’t do anything on the next paragraph start and then does whatever it wants to do, e.g., in an `\item` of a list it will typeset the label in front of the paragraph text. However, there is only one such `\everypar` token register and if different packages and/or the kernel all attempt to add their own code here, coordination is very difficult if not impossible.

The process when the paragraph ends has different mechanisms and interfaces. A paragraph ends when the engine primitive `\par` is called while T_EX is in unrestricted horizontal mode, i.e., is building a paragraph. At other times this primitive does nothing or generates as an error depending on the mode T_EX is in, e.g., the `\par` in `\hbox{a\par b}` is ignored, but `$a\par b$` would complain.

²³Already not quite true: the command `\noindent` starts the paragraph but influences the special processing by suppressing the paragraph indentation box normally inserted by it.

²⁴That’s a bit different from placing a zero-sized box!

If this primitive ends the paragraph it does some special “end of horizontal list” processing, then calls \TeX ’s paragraph builder; this breaks the horizontal list into lines and then these lines are added as boxes to the enclosing vertical list and \TeX returns to vertical mode.

This \par command can be given explicitly, but there are also situations in which \TeX is generating it on the fly. Most often this happens when \TeX encounters a blank line which is automatically changed to a \par command which is then executed. The other possibility is that \TeX encounters a command which is incompatible with horizontal processing, e.g., \vskip (a request for adding vertical space). In such cases it silently backs up, and inserts a \par in the hope that this gets it out of horizontal mode and makes the vertical command acceptable.

The important point to note here is that \TeX really inserts the command with the name \par , which can be redefined. Thus, it may not have its original “primitive” meaning and therefore may not end the horizontal list and call the paragraph builder. This approach offers some flexibility but also allows you to easily produce a \TeX document that loops forever, for example, the simple line

```
A \let\par\relax \vskip
```

will start a horizontal list at A, redefines \par , then sees \vskip and inserts \par to end the paragraph. But this now only runs \relax so nothing changes and \vskip is read again, issues a \par which In short, it only takes a plain \TeX document with five tokens to run forever (since no memory is consumed and therefore eventually exhausted).

There is no way other than changing \par to gain control at the end of a paragraph, i.e., there is no token list like \everypar that is inserted. Hence the only way to change the default behavior is to modify the action that \par executes, with similar issues as outlined before: different processes need to ensure that they do not overwrite their modifications or worse, think that the \par in front of them is the engine primitive while in fact it has already been changed by other code.

To make matters slightly worse there are a few places where \TeX handles the situation differently (most likely for speed reasons back when computers were much slower). If \TeX finds itself in unrestricted horizontal mode at the end of building a vertical box (for an \insert , \vadjust or executing the output routine code), it will finish the horizontal list not by issuing a \par command (which would be consistent with all other places) but by simply executing the primitive meaning of \par , regardless of the actual definition that \par has at the time.

Thus, if you have carefully crafted a redefined \par to execute some special actions at the end of a paragraph and you write something like

```
\vbox{Some paragraph ... text.}
```

you will find that your code does not get run for the last paragraph in that box. \LaTeX avoids this problem, by making sure that its boxes (such as \parbox or the \minipage environment, etc.) all internally add an explicit \par at the end so that such code is run and \TeX finds itself in vertical mode already without the need to start up the paragraph builder internally. But, of course, this only works for boxes under direct control of the \LaTeX kernel; if some package uses low-level \vbox es without adding this precaution the \TeX optimization kicks in and no special \par code is executed.

And there is another optimization that is painful: if a paragraph is interrupted by a mathematical display, e.g., $\[...]$ in \LaTeX or $$$...$$$ in plain \TeX , then \TeX will resume horizontal mode afterward, i.e., it will start to build a new horizontal list

without inserting an indentation box or `\everypar` at that point. However, if that list immediately ends with an explicit or implicit `\par` then TeX will simply throw away this “null” paragraph and not do its usual “end of horizontal list” processing, so this special case also needs to be accounted for when introducing any extended processing.

2 The new mechanism implemented for L^AT_EX

To improve the situation (and also to support automatic tagging of PDF documents) we now offer public as well as private hooks at the start and end of the paragraph processing. The public hooks can be used by packages (or by the user in the preamble or within the document) and using the hook mechanisms it is possible to reorder or arrange code from different packages in such a way that these can safely coexist.

To make that happen we have to make use of the basic functionality that is offered by TeX, e.g., we install special code inside `\everypar` to provide hooks at the beginning and we redefine `\par` to do some special processing when appropriate to install hooks at the end of the paragraph.

In order to make this work, we have to ensure that package use of `\everypar` is not overwriting our code. This is done through a trick: we basically hide the real `\everypar` from the packages and offer them a new token register (with the same name). So if they install their own code it doesn’t overwrite ours. Our code then inserts the new `\everypar` at the right place inside the process so that it looks as if it was the primitive `\everypar`.²⁵

At the end of the paragraph it would be great if we could use a similar trick. However, due to the fact that TeX inserts the token `\par` (that doesn’t have a defined meaning) we can’t hide “the real thing™” and offer the package an indistinguishable alternate.

Fortunately, L^AT_EX has already redefined `\par` for its own purposes. As a result there aren’t many packages that attempt to change `\par`, because without a lot of extra care that would fail miserably. But the bottom line is that, if you load a package that alters `\par` then the end of paragraph hooks are most likely not executing while that redefinition is active.²⁶

²⁵Ideally, `\everypar` wouldn’t be used at all by packages and instead they would simply write their code into the hooks now offered by the kernel. However, while this is the longterm goal and clearly an improvement (because then the packages do no longer need to worry about getting their code overwritten or needing to account for already existing code in `\everypar`), this will not happen overnight. For that reason support for this legacy method is retained.

²⁶Similarly to the `\everypar` situation, the remedy is that such packages stop doing this and instead add their alterations into the paragraph hooks now provided.

2.1 The provided hooks

para/before

para/begin

para/end

para/after

The following four public hooks are defined and executed for each paragraph:

para/before This hook is executed after the kernel hook `\@kernel@before@para@before` (discussed below) in vertical mode immediately after `\TeX` has contributed `\parskip` to the vertical list and before the actual paragraph processing in horizontal mode starts.

This hook should either not produce any typeset material or add only vertical material. If it starts a paragraph an error is generated. The reason is that we are in the starting process of processing a paragraph and so this would lead to endless recursion.²⁷

para/begin This hook is executed after the kernel hook `\@kernel@before@para@begin` (discussed below) in horizontal mode immediately before the indentation box is placed (if there is any, i.e., if the paragraph hasn't been started with `\noindent`).

The indentation box to be typeset is available to the hook as `\IndentBox` and its automatic placement (after the hook is executed) can be prevented through `\OmitIndent`. More precisely `\OmitIndent` voids the box.

The indentation box is then typeset directly after the hook execution by something equivalent to `\box\IndentBox` followed by the current content of the token register `\everypar` that it is available to the kernel or to packages (that run some legacy code).

One has to be careful not to add any code to the hook that starts its own paragraph (e.g., by adding a `\parbox` or a `\marginpar` inside) because that would call the hook inside again (as a new paragraph is started there) and thus lead to an endless recursion ending only after exhausting the available memory. This can only be done by making sure that is not executed for the inner paragraphs (or at least not recursively forever).

para/end This hook is executed at the end of a paragraph when `\TeX` is ready to return to vertical mode and after it has removed the last horizontal glue (but not any kerns) placed on the horizontal list. The code is still executed in horizontal mode so it is possible to add further horizontal material at this point, but it should not alter the mode (even a temporary exit from horizontal mode would create chaos—any attempt will cause an error message)! After the hook has ended the kernel hook `\@kernel@after@para@end` is executed and then `\TeX` returns to vertical mode.

The hook is offered as public hook, but because of the requirement to stay within horizontal mode one needs to be careful in what is placed into the hook.²⁸

This hook is implemented as a reversed hook.

para/after This hook is executed directly after `\TeX` has returned to vertical mode and after any material that migrated out of the horizontal list (e.g., from a `\vadjust`) has processed.

²⁷One could allow it but only if the newly started paragraph is processed without any hooks. Furthermore correct spacing would be a bit of a nightmare so for now this is forbidden.

²⁸Maybe we should guard against that, but it would be rather tricky to implement as mode changes can happen across group boundaries so one would need to keep a private stack just for that. Well, something to ponder.

This hook should either not produce any typeset material or add only vertical material. However, for this hook starting a new paragraph is not a disaster so that it isn't prevented.

This hook is implemented as a reversed hook.

Once that hook code has been processed the kernel hook `\@kernel@after@para@after` is executed as the final action of the paragraph processing.

```
\@kernel@before@para@before
\@kernel@after@para@after
\@kernel@before@para@begin
\@kernel@after@para@end
```

As already mentioned above there are also four kernel hooks that are executed at the start and end of the processing.

`\@kernel@before@para@before` For future extensions, not currently used by the kernel.

`\@kernel@after@para@after` For future extensions, not currently used by the kernel.

`\@kernel@before@para@begin` Used by the kernel to implement tagging. This hook is executed at the very beginning of a paragraph after TeX has switched to horizontal mode but before any indentation box got added or any `\everypar` was run.

It should not generate typeset material that could alter the position. Note that it should never leave hmode, otherwise you will end with a loop! We could guard against this, but since it is an internal kernel hook that shouldn't be touched this isn't checked.

`\@kernel@after@para@end` Used by the kernel to implement tagging. It is executed directly after the public `para/end` hook. After it there is a quick check that we are still in horizontal mode, i.e., that the public hook has not mistakenly ended horizontal mode prematurely (this is an incomplete check just testing the mode and could perhaps be improved (at the cost of speed)).

2.2 Altered and newly provided commands

```
\par
\endgraf
\para_end:
```

An explicit request for ending a paragraph is provided in plain TeX under the name `\endgraf`, which simply uses the primitive meaning (regardless of what `\par` may have as its current definition). In L^AT_EX `\endgraf` (with that behavior) was originally also available.

With the new paragraph handling in L^AT_EX, ending a paragraph means a bit more than just calling the engine's paragraph builder: the process also has to add any hook code for the end of a paragraph. Thus `\endgraf` was changed to provide this additional functionality (along with `\par` remaining subject to its current meaning).

The expl3 name for this functionality is `\para_end:`.

Note: *The next two commands are still under discussion and may slightly change their semantics (as described in the document) and/or their names between now and the 2021 Spring release!*

`\OmitIndent`
`\para omit indent:`

Inside the `para/begin` hook one can use this command to suppress the indentation box at the start of the paragraph. (Technically it is possible to use this command outside the hook as well, but this should not be relied upon.) The box itself remains available for use.

The `expl3` name for the function is `\para omit indent:`.

`\IndentBox`
`\g para indent box`

The box register holding the indentation box for the paragraph is available for inspection (or changes) inside hooks. It remains available even if the `\OmitIndent` command was used; in that case it will just not be automatically placed.

The `expl3` name for the box register is `\g para indent box`.

`\RawIndent`
`\para raw indent:`
`\RawNoindent`
`\para raw noindent:`
`\RawParEnd`
`\para raw end:`

```
\RawIndent hmode material \RawParEnd
\RawNoindent hmode material \RawParEnd
```

The commands `\RawIndent` and `\RawNoindent` are not meant for normal paragraph building (where the result is a textual paragraph in the traditional meaning of the word), but for special cases where `TeX`'s low-level algorithm is used to achieve special effects, but where the result is not a “paragraph”.

They are called “raw”, because they bypass `LATEX`'s hook mechanism for paragraphs and simply invoke the low-level `TeX` algorithm. I.e., they are like the original `TeX` primitives `\indent` and `\noindent` (that is they execute no hooks other than `\everypar`) except that they can only be used in vertical mode and generate an error if found elsewhere.

To avoid issues a paragraph started by them should always be ended by `\RawParEnd`²⁹ and not by `\par` (or a blank line), because the latter will execute hooks which then have no counterpart at the beginning of the paragraph. It is the responsibility of the programmer to make sure that they are properly paired. This also means that one should not put arbitrary user content between these commands if that content could contain stray `\pars`.

The `expl3` names for the functions are `\para raw indent:`, `\para raw indent:` and `\para raw end:`.

2.3 Examples

None of the examples in this section are meant for real use as they are far too simple-minded but they should give some ideas of what could be possible if a bit more care is applied.

2.3.1 Testing the mechanism

The idea is to output for each paragraph encountered some information: a paragraph sequence number, a level number in roman numerals, the environment in which this paragraph appears, and the line number where the start or end of the paragraph is, e.g., something like

²⁹Technical note for those who know their `TeXbook`: the `\RawParEnd` command invokes the original `TeX` engine definition of `\par` that (solely) triggers the paragraph builder in `TeX` when found inside unrestricted horizontal mode and does nothing in other processing modes.

```

PARA: 1-i start (document env. on input line 38)
PARA: 1-i end   (document env. on input line 38)
PARA: 2-i start (document env. on input line 40)
PARA: 3-ii start (minipage env. on input line 40)
PARA: 3-ii end   (minipage env. on input line 40)
PARA: 2-i end   (document env. on input line 41)

```

As you can see paragraph 2 starts on line 40 and ends on 41 and inside a minipage started paragraph 3 (start and end on line 40). If you run this on some document you will find that L^AT_EX considers more things “a paragraph” than you have probably thought.

This was generated by the following hook code:

```

\newcounter{paracnt}          % sequence counter
\newcounter{paralevel}        % level counter

```

To support paragraph nesting we need to maintain a stack of the sequence numbers. This is most easily done using `expl3` functions, so we switch over. This is not a very general implementation, just enough for what we need and a bit of L^AT_EX 2 _{ε} thrown in as well. When popping, the result gets stored in `\paracntvalue` and the `\ERROR` should never happen because it means we have tried to pop from an empty stack.

```

\ExplSyntaxOn
\seq_new:N \g_para_seq
\cs_new:Npn \ParaPush
  {\seq_gpush:No \g_para_seq {\the\value{paracnt}}}
\cs_new:Npn \ParaPop  {\seq_gpop:NNF \g_para_seq \paracntvalue \ERROR }
\ExplSyntaxOff

```

At the start of the paragraph increment both sequence counter and level and also save the then current sequence number on our stack.

```

\AddToHook{para/begin}{%
  \stepcounter{paracnt}\stepcounter{paralevel}%
  \ParaPush
}

```

To display the sequence number we `\typeout` the current sequence and level number. The command `\@currenvir` gives us the current environment and `\on@line` produces a space and the current input line number.

```

\typeout{PARA: \arabic{paracnt}-\roman{paralevel} start
(\@currenvir\space env.\on@line)}%

```

We also typeset the sequence number as a tiny red number in a box that takes up no horizontal space. This helps us seeing where L^AT_EX sees the start and end of the paragraphs in the document.

```

\llap{\color{red}\tiny\arabic{paracnt}\ }%
}

```

At the end of the paragraph we display sequence number and level again. The level counter has the correct value but we need to retrieve the right sequence value by popping it off the stack after which it is available in `\paracntvalue` the way we have set this up above.

```
\AddToHook{para/end}{%
  \ParaPop
  \typeout{PARA: \paracntvalue-\roman{paralevel} end \space\space
  (@currenvir\space env.\on@line)}%
```

We also typeset again a tiny red number with that value, this time sticking out to the right.³⁰ We also decrement the level counter since our level has finished.

```
\rlap{\color{red}\tiny\ \paracntvalue}%
\addtocounter{paralevel}{-1}%
}
\makeatother
```

2.3.2 Mark the first paragraph of each `itemize`

The code for this is rather simple. We supply some code that is executed only once inside a hook at the start of each `itemize`. We explicitly change the color back and forth so that we don't introduce grouping around the paragraph.

```
\AddToHook{env/itemize/begin}{%
  \AddToHookNext{para/begin}{\color{blue}}%
  \AddToHookNext{para/end}{\color{black}}%
}
```

As a result the first paragraph of each `itemize` will appear in blue.

2.4 Some technical notes

The code tries hard to be transparent for package code, but of course any change means that there is a potential for breaking other code. So in section we collect a few cases that may be of importance if low-level code is dealing with paragraphs that are now behaving slightly differently. The notes are from issues we observed and will probably grow over time.

2.4.1 Glue items between paragraphs (found with `fancypar`)

In the past L^AT_EX placed two glue items between two consecutive paragraphs, e.g.,

```
text1 \par text2 \par
```

would show something like

```
\glue(\parskip) 0.0 plus 1.0
\glue(\baselineskip) 5.16669
```

but now there is another `\parskip` glue (that is always 0pt):

```
\glue(\parskip) 0.0 plus 1.0
\glue(\parskip) 0.0
\glue(\baselineskip) 5.16669
```

³⁰Note that this can alter the document pagination, because a paragraph ending in a display (e.g., an equation) will get an extra line—in that case our tiny number has an effect even though it doesn't take up any space, because it paragraph is no longer empty and thus isn't dropped!

The reason is that we generate a “fake” paragraph to gain control and safely add the early hooks, but this generates an additional glue item. That item doesn’t contribute anything vertically but if somebody writes code that unravels a constructed list using `\lastbox`, `\unskip` and `\unpenalty` then the code has to remove one additional glue item or else it will fail.

3 The Implementation

```

1  <@=para>
2  <*ekernel | latexrelease>
3  \ExplSyntaxOn
4  <latexrelease> \NewModuleRelease{2021/06/01}{ltpara}
5  <latexrelease>           {Paragraph-handling-and-hooks}

```

3.1 Providing hooks for paragraphs

`para/before` The public hooks. They are implemented as a paired set of hooks.

```

para/after
para/begin
para/end

```

(*End of definition for para/before and others. These functions are documented on page 404.*)

```

@kernel@before@para@before
@kernel@after@para@after
@kernel@before@para@begin
@kernel@after@para@end

```

The corresponding kernel hooks (for tagging and future extensions).

```

8 \let \@kernel@before@para@before \empty
9 \let \@kernel@before@para@begin \empty
10 \let \@kernel@after@para@end \empty
11 \let \@kernel@after@para@after \empty

```

(*End of definition for @kernel@before@para@before and others. These functions are documented on page 405.*)

```
\g_para_standard_everypar_t1
```

Whenever TeX starts a paragraph it inserts first an indentation box and then executes the tokens stored in `\tex_everypar:D` (known to L^AT_EX as `\everypar`). We alter this behavior slightly here, so that hooks are added into the right place. Otherwise the process change remains transparent to any legacy code for this space.

We keep the standard code to be used by `\tex_everypar:D` in a separate token list because we have to switch back and forth for error recovery and so altering `\tex_everypar:D` all the time should be a tiny bit faster.

```

12 <latexrelease> \IncludeInRelease{2023/06/01}
13 <latexrelease>   {\g_para_standard_everypar_t1}{minipage- fix}
14 \tl_new:N \g_para_standard_everypar_t1

```

Here is now its definition:

```
15 \tl_gset:Nn \g_para_standard_everypar_t1 {
```

First we remove the indentation box and store it in `\g_para_indent_box`. If there was none because the paragraph was started by `\noindent` the box register will be void.

```
16 \box_gset_to_last:N \g_para_indent_box
```

This will make the newly started horizontal list empty, so if we stop it now and return to vertical mode it will be dropped by TeX. We do that but inside a group so that any `\parshape` settings will not get lost as we need them for later.

```

17 \group_begin:
18   \tex_par:D
19 \group_end:

```

We then change `\tex_everypar:D` to generate an error so that we can detect and report if the `para/before` hook illegally changed out of vmode.

```
20  \tex_everypar:D { \msg_error:nnn { hooks }{ para-mode }{before}{vertical} }
21  \@kernel@before@para@before
22  \hook_use:n {para/before}
```

Assuming the hooks have been well behaved it is time to return to horizontal mode and start the paragraph in earnest. We already have the indentation box saved away so we now have to restart the paragraph with an empty `\tex_everypar:D` and with `\tex_noindent:D`. And we need to make sure not to get another `\parskip` or rather (since we can't prevent that) that it is of zero size.

```
23  \group_begin:
24  \tex_everypar:D {}
```

There has been a long-standing problem with L^AT_EX's minipages in that invisible material at the beginning of a minipage (such as a `\color` setting) would result in `\parskip` being added in front of the first paragraph—something that is not done by T_EX if a vertical list is completely empty. As this is happening on a very low-level in the engine it wasn't really possible to find out if this `\parskip` was added or if a space we see in front of the current point is legitimate. However, with the new paragraph handling we are in a better position: while we still don't know if there is such a space or not, we do know if we have just created an empty paragraph. Thus, if we now set `\parskip` to `-\parskip` the two will cancel each other if present and if the first was ignored because the vertical list was empty, then the second will be ignored too because it is still empty. Of course, we don't want to cancel always but only at the start of a minipage and that is signaled with the `@minipage` switch.

```
25  \skip_set:Nn \tex_parskip:D
26  { \if@minipage -\tex_parskip:D \else: \c_zero_skip \fi: }
27  \tex_noindent:D
28  \group_end:
```

That brings us back to the start of the horizontal list but we need to change `\tex_everypar:D` back to its normal content in case there are nested paragraphs coming up.

```
29  \tex_everypar:D{\g__para_standard_everypar_t1}
```

This is followed by executing the kernel and the public hook. The kernel hook is there to enable tagging.

```
30  \@kernel@before@para@begin
31  \hook_use:n {para/begin}
```

If we aren't in horizontal mode any longer the hooks above misbehaved.

```
32  \if_mode_horizontal: \else:
33  \msg_error:nnn { hooks }{ para-mode }{begin}{horizontal} \fi:
```

Finally we reinsert the indentation box (unless suppressed) and then call `\everypar` the way legacy L^AT_EX code expects it.

However, adding the public `\everypar` is a bit tricky (see below) so we add that later, and indirectly.

```
34  \__para_handle_indent:
35  % \the \everypar           % <--- done differently below
36  }
```

```

37  ⟨latexrelease⟩\cs_set:Npn \__para_tmp:w #1#2#3#4#5 { }
38  ⟨latexrelease⟩\tl_gput_right:Nx \g__para_standard_everypar_tl {
39  ⟨latexrelease⟩    \exp_not:N \the
40  ⟨latexrelease⟩    \exp_not:N \toks
41  ⟨latexrelease⟩    \exp_after:wN \__para_tmp:w \token_to_meaning:N \everypar
42  ⟨latexrelease⟩    \c_space_tl
43  ⟨latexrelease⟩}
44  ⟨latexrelease⟩\EndIncludeInRelease
45  ⟨latexrelease⟩\IncludeInRelease{2021/06/01}
46  ⟨latexrelease⟩      {\g__para_standard_everypar_tl}{minipage~ fix}
47  ⟨latexrelease⟩
48  ⟨latexrelease⟩\tl_gset:Nn \g__para_standard_everypar_tl {
49  ⟨latexrelease⟩  \box_gset_to_last:N \g_para_indent_box
50  ⟨latexrelease⟩  \group_begin:
51  ⟨latexrelease⟩    \tex_par:D
52  ⟨latexrelease⟩  \group_end:
53  ⟨latexrelease⟩  \tex_everypar:D {\msg_error:nnnn {hooks }{para-mode }{before}{vertical} }
54  ⟨latexrelease⟩  \@kernel@before@para@before
55  ⟨latexrelease⟩  \hook_use:n {para/before}
56  ⟨latexrelease⟩  \group_begin:
57  ⟨latexrelease⟩    \tex_everypar:D {}
58  ⟨latexrelease⟩    \skip_zero:N \tex_parskip:D
59  ⟨latexrelease⟩    \tex_noindent:D
60  ⟨latexrelease⟩  \group_end:
61  ⟨latexrelease⟩  \tex_everypar:D{\g__para_standard_everypar_tl}
62  ⟨latexrelease⟩  \@kernel@before@para@begin
63  ⟨latexrelease⟩  \hook_use:n {para/begin}
64  ⟨latexrelease⟩  \if_mode_horizontal: \else:
65  ⟨latexrelease⟩    \msg_error:nnnn {hooks }{para-mode }{begin}{horizontal} \fi:
66  ⟨latexrelease⟩  \__para_handle_indent:
67  ⟨latexrelease⟩}

```

We also have to add the `\everypar` toks register at the end. In case of rollback this is already allocated and we have to find out the correct number (hope this is correctly done)

```

68  ⟨latexrelease⟩\cs_set:Npn \__para_tmp:w #1#2#3#4#5 { }
69  ⟨latexrelease⟩\tl_gput_right:Nx \g__para_standard_everypar_tl {
70  ⟨latexrelease⟩    \exp_not:N \the
71  ⟨latexrelease⟩    \exp_not:N \toks
72  ⟨latexrelease⟩    \exp_after:wN \__para_tmp:w \token_to_meaning:N \everypar
73  ⟨latexrelease⟩    \c_space_tl
74  ⟨latexrelease⟩}
75  ⟨latexrelease⟩\EndIncludeInRelease

```

(End of definition for `\g__para_standard_everypar_tl`.)

`\tex_everypar:D` `\tex_everypar:D` then only has to execute `\g__para_standard_everypar_tl` by default.

```

76  \tex_everypar:D{\g__para_standard_everypar_tl}

```

(End of definition for `\tex_everypar:D`.)

`\everypar` Tokens inserted at the beginning of the paragraph are placed into `\everypar` inside legacy L^AT_EX code, e.g., by the list environments or by headings to handle `\clubpenalty`, etc.

Now this isn't any longer the primitive but simply a toks register used in the code above but to legacy L^AT_EX code that is transparent.

There is, however, a problem: a handful packages use exactly the same trick and replace the primitive with a token register and call the token register inside the renamed primitive. That is they assume that `\everypar` is the primitive and that it will still be called at the start of the paragraph even if renamed.

But if we have already replaced it by a token register then all they do is to give that token register a new name. Thus our code in `\tex_everypar:D` would call `\everypar` (which is now their token register) and the code that they added ends up in our token register which is then never used at all. A bit mind boggling I guess.

So what we have to do is not to call the token register `\everypar` by its name inside `\tex_everypar:D` but by using its actual register number.

77 `\newtoks \everypar`

After we have allocated a new toks register with the name `\everypar` the actual register number is available (briefly) inside `\allocationnumber`. So instead of `\the\everypar` we have to put `\the\toks<allocated number>` at the end of `\tex_everypar:D`.

So what remains doing is to append a few tokens to the token list `\g_para-standard_everypar_tl` which we do now. We use x expansion here to get the value of `\allocationnumber` in, all the other tokens should not be expanded at this point.

One important point here is to terminate the register allocation number with a real space. This space will get swallowed up when the number is read. Anything else, such as `\scan_stop:` would remain in the input and that would mean that it would interfere with `\everypar` code that attempts to scan ahead to see how the paragraph text starts.

78 `\t1_gput_right:Nx \g_para_standard_everypar_tl {`
79 `\exp_not:N \the`
80 `\exp_not:N \toks`
81 `\the \allocationnumber`
82 `\c_space_tl`
83 `}`

(End of definition for `\everypar`.)

`\g_para_indent_box` For managing the indentation we need to provide a public accessible box register

84 `\box_new:N \g_para_indent_box`

(End of definition for `\g_para_indent_box`. This function is documented on page 406.)

`_para_handle_indent:` Adding (typesetting) the indent box is straight forward. If it was emptied before it does nothing.

85 `\cs_new:Npn _para_handle_indent: {`
86 `\box_use_drop:N \g_para_indent_box`
87 `}`

The declaration `\para omit indent:` (or `\OmitIndent`) changes that to do nothing.

88 `\cs_new:Npn \para omit indent: {`
89 `\box_gclear:N \g_para_indent_box`
90 `}`

(End of definition for `_para_handle_indent:..`)

`\IndentBox` The L^AT_EX 2 _{ε} names for the indentation box and for suppressing it for use in the `para/begin` hook.

```
91 \cs_set_eq:NN \IndentBox \g_para_indent_box  
92 \cs_set_eq:NN \OmitIndent \para omit indent:
```

(End of definition for `\IndentBox` and `\OmitIndent`. These functions are documented on page 406.)

`\para_end:` Adding hooks to the end of a paragraph is similar but here we need to alter the command that is used by T_EX to end horizontal mode and return to vertical mode, i.e., `\par`.

This is a bit more complicated as this command can appear anywhere either explicitly or implicitly added by T_EX in certain situations:

- when using `\par` in the code or the document
- when using a blank line (which is converted to `\par`)
- when T_EX finds any commands incompatible with horizontal mode it issues a `\par` and then rereads the command.

Unfortunately, T_EX has some (these days) unnecessary optimizations: if a `\vbox` ends and T_EX is still in horizontal mode it simply exercises the paragraph builder instead of issuing a `\par`. It is therefore necessary for L^AT_EX to ensure that this case doesn't happen and all boxes internally have a `\par` command at their end.

This `\par` may or may not run the “par primitive” (which is always available as `\tex_par:D` in `expl3`); it is permissible to have a changed meaning and it is in fact changed by L^AT_EX in various ways at various points inside `latex.ltx`. For this L^AT_EX 2 _{ε} code has the following conventions: `\@@par` and `\endgraf` both refer to the default meaning (in the past this was the `initex` primitive) while `\par` is the current meaning which maybe does something else.

We are now going to change this default meaning to instead run `\para_end:`, which ultimately executes the `initex` primitive but additionally adds our hooks when appropriate. This way the change is again transparent to the legacy L^AT_EX 2 _{ε} code.

In most cases `\para_end:` should behave exactly like the primitive and we achieve this by simply expanding it to the primitive which is available to us as `\tex_par:D`. This way we don't have to care about whether T_EX just does nothing (e.g., if in vertical mode already) or generates an error, etc.

```
93 \cs_new_protected:Npn \para_end: {
```

CCC Maybe needs more explanation. TEMP NOTE: What should happen if in outer hmode with an empty hlist?

The only case we care about is when we are in horizontal mode (i.e., doing typesetting) and not also in inner mode (i.e., making paragraphs and not building an `\hbox`).

```
\bool_lazy_and:nnT  
  { \mode_if_horizontal_p: }  
  { \bool_not_p:n { \mode_if_inner_p: } }  
  { ... }
```

Since this is executed for each and every paragraph in a document we try to stay as fast as possible, so we do not use the above construct but two conditionals instead. Using low-level `\if_mode...` conditions would be even faster but has the danger to conflict with conditionals in the user hooks.

If `\para_end:` is executed while T_EX is currently doing a low-level assignment the test for horizontal mode may get executed as part of the assignment. That is normally not an issue but we just found one case where it is:

```
\afterassignment\lst@vskip\@tempskipa \z@ \par
```

If \TeX is in hmode while that assignment happens then the \par is seen in hmode because in the above case the assignment may not be finished (one should have used \z@skip) and the \lst@vskip will get inserted into the middle of the conditional. The \lst@vskip then changes to vmode and you get a surprising error about the para/end hook having changed modes even if you don't have any hook code(!): it is the inserted \lst@vskip that is actually causing the change of mode. This is what happened when the output routines got started while a lstlisting environment (that redefines \vskip in this way) was active. This is really faulty coding, but we try to be proactive and guard the conditional so that any scanning is first stopped, thus:

```
94   \scan_stop:  
95   \mode_if_horizontal:TF {  
96     \mode_if_inner:F {
```

In that case the action of the primitive would be to remove the last glue (but no kerns) from the horizontal list (constructed to form a paragraph) and then to append a penalty of 10000 and the \parfillskip ; it then passes the whole list to the paragraph builder, which breaks it into lines and \TeX then returns to vertical mode.

What we want to do is to add this hook code at the end of the horizontal list before any of the above happens. If there was a glue item at the end of the list then it should get removed before the hook code gets added so we have to arrange for this removal.

As in other similar cases, it may be best to add here a \nobreak in case the hook itself adds glue and thus creates a non-explicit and unwanted potential break point. On the other hand (as has been argued) the code in the hook should perhaps have the responsibility for adding such a guard penalty in this case. This needs further analysis and decisions (as in emails).

In either case, good documentation of these hooks is essential, covering what the hook may or should provide and all such related considerations concerning the content.

There is not much point in checking if there was really a glue item at the end of the horizontal list, instead we simply try to remove one using \tex_untkip:D : if there wasn't one this will do nothing.

```
97   \tex_untkip:D
```

We then execute the public hook (which may add some final typeset material) followed by the kernel hook that we need for adding tagging support. None of this is supposed to change the mode—at the moment we make only a very simple test for this, more devious changes go unnoticed, but too bad as they will then probably backfire badly.

```
98   \hook_use:n{para/end}  
99   \@kernel@after@para@end  
100  \mode_if_horizontal:TF {
```

The final action (before getting to the point where \tex_par:D is called) is to add an extra glue item so that the primitive is prevented from removing intended glue (if there was some). If we don't do this and the horizontal list ends in several glue items we would end up removing two glue items instead of just the last one, which would be wrong. We use glue (rather than a kern) as that will be removed by the primitive.

There is however one other \TeX optimization that hurts: in a sequence like this $\$\dots \\par (with \par being the primitive) \TeX will be in horizontal mode after the display, ready to receive further paragraph text, but since the \par follows immediately there is a “null” paragraph at the end and \TeX simply throws that away. The space between $\$\dots \$$ and \par got already dropped during the display processing so the \par is

not removing any space and appending `\parfillskip`, instead it simply goes silently to vmode.

Now if we would have added something (to prevent glue removal) that would look to TeX like material after the display and so we would end up with an empty paragraph just containing a penalty and `\parfillskip`.

We therefore check if the current hlist does end in glue (`\tex_lastnodetype:D` has the value 11) and if so we add a zero-length guard skip which will be removed by the following `\tex_par:D`.

```
101   \if_int_compare:w 11 = \tex_lastnodetype:D
102     \tex_hskip:D \c_zero_dim
103   \fi:
```

To run the `para/after` hook we first end the paragraph. This means that the `\tex_par:D` at the very end is unnecessary but executing it there unnecessarily is better than having code that tests for all the different mode possibilities.

```
104   \tex_par:D
105   \hook_use:n{para/after}
106   \o@kernel@after@para@after
107 }
```

If we were not horizontal mode (the F case from above) then the earlier hook `para/end` must have been at fault, so we report that.

```
108   { \msg_error:nnn { hooks }{ para-mode }{end}{horizontal} }
```

Finally close out the nested conditionals.

```
109 }
110 }
```

And then we can use the primitive to truly end the paragraph.

```
111 \tex_par:D
112 }
```

(End of definition for `\para_end`. This function is documented on page 405.)

`\para_raw_indent:` and `\para_raw_noindent:` are like the primitives `\indent` and `\noindent` except that they can only be used in vertical mode.

To avoid issues a paragraph started by them should always be ended by `\para_raw_end:` and not by `\para_end:` or `\par` as the latter will execute hooks which then have no counterpart at the beginning of the paragraph. It is the responsibility of the programmer to make sure that they are properly paired.

```
113 \cs_new:Npn \para_raw_indent: {
114   \mode_if_vertical:TF
115   {
116     \tex_everypar:D {
117       \box_gset_to_last:N \g_para_indent_box
118       \tex_everypar:D { \g__para_standard_everypar_tl }
119       \__para_handle_indent:
120       \the\tex_everypar }
121     }
122     { \msg_error:nn { latex2e }{ raw-para } }
123   \tex_indent:D
124 }
```

```

125 \cs_new:Npn \para_raw_noindent: {
126   \mode_if_vertical:TF
127   {
128     \tex_everypar:D {
129       \tex_everypar:D { \g__para_standard_everypar_tl }
130       \the\tex_everypar }
131     }
132     { \msg_error:nn { latex2e }{ raw-para } }
133   \tex_noindent:D
134 }
135 \cs_new_eq:NN \para_raw_end: \tex_par:D

```

(End of definition for `\para_raw_indent:`, `\para_raw_noindent:`, and `\para_raw_end:`. These functions are documented on page 406.)

\RawIndent The L^AT_EX 2_E names for starting and ending a paragraph without adding any hooks.

```

136 \cs_set_eq:NN \RawIndent \para_raw_indent:
137 \cs_set_eq:NN \RawNoIndent \para_raw_noindent:
138 \cs_set_eq:NN \RawParEnd \para_raw_end:

```

(End of definition for `\RawIndent`, `\RawNoIndent`, and `\RawParEnd`. These functions are documented on page 406.)

This ends the `para` module code.

```
139 {@@=}
```

\par Having the new default definition for `\par` we also have to set it up so that it gets used. This involves three commands: `\par`, `\@@par` (to which L^AT_EX resets `\par` occasionally) and `\endgraf`, which is another name for the “default” action of `\par`.

```

140 \cs_set_eq:NN \par \para_end:
141 \cs_set_eq:NN \@@par \para_end:
142 \cs_set_eq:NN \endgraf \para_end:

```

(End of definition for `\par`, `\endgraf`, and `\@@par`. These functions are documented on page 405.)

While this is not integrated properly into the format we have to redo the `\everypar` setting from the kernel, otherwise that gets lost (as it happens before that file is loaded).

```
143 \everypar{\@nodocument} %% To get an error if text appears before the \document
```

3.2 The error messages

This one is used when we detect that some hook code has changed the mode where it shouldn’t, e.g., by starting or ending a paragraph. The first argument is the hook name second the mode it should have stayed in but didn’t.

```

144 \msg_new:nnnn { hooks } { para-mode }
145   {
146     Illegal~mode~ change~ in~ hook~ 'para/#1'.\\
147     Hook~ code~ did~ not~ remain~ in~ #2~ mode.
148   }
149   {
150     Paragraph~ hooks~ cannot~ change~ the~ TeX~ mode~ without~ causing~
151     endless~ recursion.~ The~ hook~ code~ in~ 'para/#1'~ needs~ to~ stay~
152     in~ #2~ mode,~ but~ it~ didn't.~ Examine~ the~ hook~
153     code~ with~ \iow_char:N \\ShowHook~ to~ find~ the~ issue.
154   }

```

And here is one used in the “raw” commands when they are used outside of vertical mode.

```

155 \msg_new:nnnn { latex2e } { raw-para }
156   {
157     Not~ in~ vertical~ mode.
158   }
159   {
160     Starting~ a~ paragraph~ with~ \iow_char:N \\RawIndent~ or~
161     \iow_char:N \\RawNoindent \\
162     (or~ \iow_char:N \\para_raw_indent:~ or~
163     \iow_char:N \\para_raw_noindent:)~ is~ only~ allowed \\
164     if~ LaTeX~ is~ in~ vertical~ mode.
165   }
166 %
167 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
168 ⟨latexrelease⟩                               {ltpara}{Undo-hooks-for-paragraphs}
169 ⟨latexrelease⟩
170 ⟨latexrelease⟩\let \OmitIndent  \undefined
171 ⟨latexrelease⟩\let \IndentBox   \undefined
172 ⟨latexrelease⟩\let \RawIndent   \undefined
173 ⟨latexrelease⟩\let \RawNoindent \undefined
174 ⟨latexrelease⟩\let \RawParEnd  \undefined
175 ⟨latexrelease⟩
176 ⟨latexrelease⟩\cs_set_eq:NN \par      \tex_par:D
177 ⟨latexrelease⟩\cs_set_eq:NN \@@par    \tex_par:D
178 ⟨latexrelease⟩\cs_set_eq:NN \endgraf \tex_par:D
179 ⟨latexrelease⟩

```

We also need to clean up the primitive “everypar” as that should no longer execute any code by default. And, of course, make `\everypar` become the primitive again.

```

180 ⟨latexrelease⟩\tex_everypar:D {}
181 ⟨latexrelease⟩\cs_set_eq:NN \everypar \tex_everypar:D
182 ⟨latexrelease⟩
183 ⟨latexrelease⟩\EndModuleRelease
184 \ExplSyntaxOff
185 ⟨/2ekernel | latexrelease⟩

```

File 17

ltmeta.dtx

Abstract

This code defines the `\DocumentMetadata` interface.

1 Introduction

In the past there was no dedicated location to declare settings concerning a document as a whole. Settings are placed somewhere in the preamble or with the class options or even with some package options. For some settings this can be too late, for example the pdf version can no longer be changed if a package has used code which already opened the PDF.

`\DocumentMetadata` as a new command unifies such settings in one place. It must be used before `\documentclass` but can be issued more than once there.

At the moment most of the code run by `\DocumentMetadata` is external to the format and subject to change. This includes the supported key/values.

For that reason all that happens right now in the format is to look for suitable support files and if found, to redirect the processing to them.

1.1 `\DocumentMetadata`

```
\DocumentMetadata \DocumentMetadata{\{key-value list\}}
```

The keys defined for `\DocumentMetadata` currently allow to set the PDF version, to set the PDF `/Lang`, to uncompress a PDF, to set the language and to declare a few PDF standards and some color profiles.

`\DocumentMetadata` is also used to activate the new PDF management code and it loads a number of required files for the PDF management code. As this forces the loading of the backend files, a backend which can't be detected automatically like `dvipdfmx`, must be set in the first `\DocumentMetadata` call (if there is more than one).

The full set of keys currently supported is documented in `documentmetadata-support.pdf` for now.

2 The Implementation

```
1  {*2ekernel | latexrelease}
Not needed yet but ...
2  %\ExplSyntaxOn
3  \let \IfDocumentMetadataTF \@secondoftwo
4  \protected\def\DocumentMetadata{%
5    \IfDocumentMetadataTF \@secondoftwo
6    \InputIfFileExists{documentmetadata-support.ltx}%
7    {}%
8  }%
```

The above file is changing `\DocumentMetadata` to a suitable definition (or so we hope), so that we can try again — if not tough.

If the file can't be found we say so and carry on without it.

```

9   {%
10    \@latex@error{No support files for
11      \noexpand\DocumentMetadata found}
12    {Is the 'LaTeX-lab' bundle installed?%
13     \MessageBreak
14     Without it, the declaration is ignored.}%

```

No point in trying this more than once if there are several calls in the document.

```

15   \let\DocumentMetadata\@gobble
16   }%
17   \let \IfDocumentMetadataTF \@firstoftwo
18   \DocumentMetadata
19 }

```

To allow package and class author to support for document links we provide also the new interface commands of the hyperref package for the creation of targets.

```

\MakeLinkTarget
  \LinkTargetOn
  \LinkTargetOff
\NextLinkTarget
20  \langle latexrelease \rangle \IncludeInRelease{2024/11/01}%
21  \langle latexrelease \rangle           {\MakeLinkTarget}{Record target name for tagging support}%
22  \ExplSyntaxOn
23  \int_new:N \g__kernel_target_int
24  \NewDocumentCommand \MakeLinkTarget{sO{}m}{%
25    \ifvmode
26      \special{}%
27    \else
28      \@savsf \spacefactor
29      \smash{}%
30      \spacefactor \@savsf
31    \fi
32    \IfBooleanTF {#1}
33    {
34      \tl_gset:Ne \currentHref {#3}
35    }
36    {
37      \int_gincr:N \g__kernel_target_int
38      \tl_gset:Ne \currentHref {target*.int_use:N \g__kernel_target_int}
39    }
40    \UseTaggingSocket{recordtarget}
41  }
42 \ExplSyntaxOff
43 \langle latexrelease \rangle \EndIncludeInRelease
44 \langle latexrelease \rangle \IncludeInRelease{2022/06/01}%
45 \langle latexrelease \rangle           {\MakeLinkTarget}{Record target name for tagging support}%
46 \langle latexrelease \rangle \NewDocumentCommand \MakeLinkTarget{sO{}m}{%
47 \langle latexrelease \rangle   \ifvmode
48 \langle latexrelease \rangle   \special{}%
49 \langle latexrelease \rangle   \else
50 \langle latexrelease \rangle   \@savsf \spacefactor
51 \langle latexrelease \rangle   \smash{}%
52 \langle latexrelease \rangle   \spacefactor \@savsf
53 \langle latexrelease \rangle   \fi}
54 \langle latexrelease \rangle \EndIncludeInRelease

```

```
55  \NewDocumentCommand\LinkTargetOn{}{}
56  \NewDocumentCommand\LinkTargetOff{}{}
57  \NewDocumentCommand\NextLinkTarget{m}{}{}
```

(End of definition for `\MakeLinkTarget` and others.)

We do not undo `\MakeLinkTarget` and friends if we roll back, in case they are used in packages that themselves do not offer rollback. This way a roll forward adds them, but the dummies remain if you roll back and you don't get missing csname errors if they are used.

```
58  \begin{macro}{\IncludeInRelease[0000/00/00]{ltmeta}}
59  % Undo Document Metadata handling
60  \end{macro}
61  \begin{macro}{\let\DocumentMetadata@\undefined}
62  \end{macro}
63  \end{macro}
```

Again for the future ...

```
64  \ExplSyntaxOff
65  \if2ekernel\else\begin{macro}{\IncludeInRelease[0000/00/00]{ltmeta}}
```

Restore module prefix (if any):

```
66  \end{macro}
```

File 18

ltspacex.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  
  \predisplaypenalty  
  \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):  
      \obeyscr : defines <CR> == \\relax  
      \restorescr : 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 `\vadjusts` and getting better error trapping (this seems to involve a lot more tokens).
- Investigate whether `\\"`, etc need to be errors in vmode; I think that they could be noops (maybe with a warning).
- Make all(?) `\unskip`s include test for zero skip (rather than other tests or no test).
- Consider replacing `\hskip 1sp` by something better (here called an ‘infinitesimal’ skip).
- Look at all `\hskip\z@` (or similar) to see if they should be changed to an ‘infinitesimal’ skip.

- Resolve the inconsistency between `\hspace` and `\hspace*`.
- Remove unnecessary `\unskip`s.
- Investigate and rationalise the ‘newline’ code.
- Find better algorithms for all sorts of things or, easier(?), fix TeX itself.

1.4 The code

```

1  {*2ekernel}
2  \message{spacing,}
3  </2ekernel>
4  {*2ekernel | latexrelease}
5  <latexrelease>\IncludeInRelease{2019/10/01}%
6  <latexrelease>           {\pagebreak}{Make commands robust}%

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

(End of definition for \pagebreak and \nopagebreak.)

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

(End of definition for \linebreak and \nolinebreak.)

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

(End of definition for \samepage.)

20 </2ekernel | latexrelease>
21 <latexrelease>\EndIncludeInRelease
22 <latexrelease>\IncludeInRelease{0000/00/00}%
23 <latexrelease>           {\pagebreak}{Make commands robust}%
24 <latexrelease>
25 <latexrelease>\kernel@make@fragile\pagebreak
26 <latexrelease>\kernel@make@fragile\nopagebreak
27 <latexrelease>\kernel@make@fragile\linebreak
28 <latexrelease>\kernel@make@fragile\nolinebreak
29 <latexrelease>\kernel@make@fragile\samepage
30 <latexrelease>
31 <latexrelease>\EndIncludeInRelease
32 {*2ekernel}

```

```

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

```

(End of definition for \@no@pgbk.)

```

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

```

(End of definition for \@no@lnbk.)

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

```

```
63 \let\\@normalcr
64 </2ekernel | latexrelease>
65 <latexrelease>\EndIncludeInRelease
66 <latexrelease>\IncludeInRelease{0000/00/00}%
67 <latexrelease>                                {\@normalcr}{\Make robust}%
68 <latexrelease>
69 <latexrelease>\DeclareRobustCommand\\{%
70 <latexrelease>    \let \reserved@c \relax
71 <latexrelease>    \let \reserved@f \relax
72 <latexrelease>    \c@ifstar{\let \reserved@c \vadjust \let \reserved@f \nobreak
73 <latexrelease>                                \cnewline}%
74 <latexrelease>    \cnewline}
75 <latexrelease>\expandafter\let\expandafter\@normalcr
76 <latexrelease>    \csname\expandafter\@gobble\string\\ \endcsname
77 <latexrelease>
78 <latexrelease>\EndIncludeInRelease
79 <*2ekernel>
```

(End of definition for \\ and \normalcr.)

`\@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.

```
80 </2ekernel>
81 <*2ekernel | latexrelease>
82 <latexrelease>\IncludeInRelease{2020/10/01}%
83 <latexrelease>                                { \vspace{0pt} }% Add calc support
84 \def\@vspace#1{\setlength{\sp@ce@skip{#1}\vskip\sp@ce@skip}%
85 </2ekernel | latexrelease>
86 <latexrelease>\EndIncludeInRelease

87 <latexrelease>\IncludeInRelease{0000/00/00}%
88 <latexrelease>                                { \vspace{0pt} }% Add calc support
89 <latexrelease>
90 <latexrelease>\let\@vspace\calcify\undefined
91 <latexrelease>\EndIncludeInRelease
92 <*2ekernel>
```

(End of definition for \vspace{\calcif{y}}.)

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

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

(End of definition for \newline.)

\@xnewline

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

(End of definition for \@xnewline.)

\@newline

```
97 </2ekernel>
98 <*2ekernel | latexrelease>
99 <latexrelease> \IncludeInRelease{2020/10/01}%
```

```

100 <{latexrelease}           {\@newline}{\newline calc support}%
101 \def\@newline[#1]{\let \reserved@e \vadjust
102                               \@gnewline {\@vspace@calcify{#1}}}
103 </2ekernel | latexrelease>
104 <{latexrelease}\EndIncludeInRelease
105 <{latexrelease}\IncludeInRelease{0000/00/00}%
106 <{latexrelease}           {\@newline}{\newline calc support}%
107 <{latexrelease}>
108 <{latexrelease}\def\@newline[#1]{\let \reserved@e \vadjust
109 <{latexrelease}           \@gnewline {\vskip #1}}
110 <{latexrelease}\EndIncludeInRelease
111 <{*2ekernel}>

```

(End of definition for `\@newline`.)

`\@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

```

112 \def\@gnewline #1{%
113   \ifvmode
114     \nolnerr
115   \else
116     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break
117   \fi}

```

(End of definition for `\@gnewline`.)

`\@getpen`

```

118 \def\@getpen#1{\ifcase #1 \z@ \or \clowpenalty\or
119   \medpenalty \or \chighpenalty
120   \else \z@\fi}

```

(End of definition for `\@getpen`.)

`\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.

```

121 \def\@nobreakfalse{\global\let\if@nobreak\iffalse}
122 \def\@nobreaktrue {\global\let\if@nobreak\iftrue}
123 \z@\nobreakfalse

```

(End of definition for `\if@nobreak`.)

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

```

124 \newdimen\@savsk
125 \newcount\@savsf

```

(End of definition for `\@savsk` and `\@savsf`.)

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

Before giving the current definition we give an extended definition that is currently not used (because it doesn't work as advertised:-)

These are generalised hacks which attempt to do sensible things when ‘invisible commands’ appear in vmode too.

They need to cope with space in both hmode (plus spacefactor) and vmode, and also cope with breaks etc. In vmode this means ensuring that any following \addvspace, etc sees the correct glue in \lastskip.

In fact, these improved versions should be used for other cases of ‘whatsits, thingies etc’ which should be invisible. They are only for commands, not environments (see notes on \@EspHack).

BTW, anyone know why the standard hacks are surrounded by \ifmmode\else rather than simply \ifhmode?

And are there any cases where saving the spacefactor is essential? I have some extensions where it is, but it does not appear to be so in the standard uses.

```
def \@bsphack{%
  \relax \ifvmode
    \@savsk \lastskip
    \ifdim \lastskip=\z@
  \else
    \vskip -\lastskip
  \fi
\else
  \ifhmode
    \@savsk \lastskip
    \@savsf \spacefactor
  \fi
\fi
```

I think that, in vmode, it is the safest to put in a \nobreak immediately after such things since writes, inserts etc followed by glue give valid breakpoints and, in general, it is possible to create breaks but impossible to destroy them.

```
def \@esphack{%
  \relax \ifvmode
    \nobreak
    \ifdim \@savsk=\z@
  \else
    \vskip\@savsk
  \fi
\else
  \ifhmode
    \spacefactor \@savsf
    \ifdim \@savsk>\z@
      \ignorespaces
    \fi
  \fi
\fi
```

```

    \fi
\fi

```

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

```

126 \def\@bsphack{%
127   \relax
128   \ifhmode
129     \csavsk\lastskip
130     \csavsf\spacefactor
131   \fi}

```

(End of definition for `\@bsphack`.)

- `\@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.

```

132 </2ekernel>
133 <latexrelease>\IncludeInRelease{2018/10/10}%
134 <latexrelease>           {\@esphack}{hyphenation and nobreak after space hack}%
135 <*2ekernel | latexrelease>
136 \def\@esphack{%
137   \relax
138   \ifhmode
139     \spacefactor\@savsf
140     \ifdim\@savsk>\z@
141       \ifdim\lastskip=\z@
142         \nobreak \hskip\z@skip
143       \fi
144       \ignorespaces
145     \fi
146   \else
147     \ifvmode
148       \if@nobreak\nobreak\else\if@noskipsec\nobreak\fi\fi
149     \fi
150   \fi}%
151 </2ekernel | latexrelease>
152 <latexrelease>\EndIncludeInRelease
153 <latexrelease>\IncludeInRelease{2015/10/01}%
154 <latexrelease>           {\@esphack}{hyphenation and nobreak after space hack}%
155 <latexrelease>\def\@esphack{%
156 <latexrelease> \relax
157 <latexrelease> \ifhmode
158 <latexrelease>   \spacefactor\@savsf
159 <latexrelease>   \ifdim\@savsk>\z@
160 <latexrelease>     \ifdim\lastskip=\z@
161 <latexrelease>       \nobreak \hskip\z@skip
162 <latexrelease>     \fi

```

```

163 〈\latexrelease〉      \ignorespaces
164 〈\latexrelease〉      \fi
165 〈\latexrelease〉  \fi}%
166 〈\latexrelease〉\EndIncludeInRelease
167 〈\latexrelease〉\IncludeInRelease{2015/01/01}%
168 〈\latexrelease〉          {\@esphack}{hyphenation and nobreak after space hack}%
169 〈\latexrelease〉\def\@esphack{%
170 〈\latexrelease〉  \relax
171 〈\latexrelease〉  \ifhmode
172 〈\latexrelease〉    \spacefactor\@savsf
173 〈\latexrelease〉    \ifdim\@savsk>\z@
174 〈\latexrelease〉      \nobreak \hskip\z@skip
175 〈\latexrelease〉      \ignorespaces
176 〈\latexrelease〉    \fi
177 〈\latexrelease〉  \fi}%
178 〈\latexrelease〉\EndIncludeInRelease
179 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
180 〈\latexrelease〉          {\@esphack}{hyphenation and nobreak after space hack}%
181 〈\latexrelease〉\def\@esphack{%
182 〈\latexrelease〉  \relax
183 〈\latexrelease〉  \ifhmode
184 〈\latexrelease〉    \spacefactor\@savsf
185 〈\latexrelease〉    \ifdim\@savsk>\z@
186 〈\latexrelease〉      \ignorespaces
187 〈\latexrelease〉    \fi
188 〈\latexrelease〉  \fi}%
189 〈\latexrelease〉\EndIncludeInRelease
190 〈*2ekernel〉

```

(End of definition for \@esphack.)

\@Eshack 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.

```

191 〈/2ekernel〉
192 〈\latexrelease〉\IncludeInRelease{2015/01/01}%
193 〈\latexrelease〉          {\@Eshack}{hyphenation after space hack}%
194 〈*2ekernel | \latexrelease〉
195 \def\@Eshack{%
196   \relax
197   \ifhmode
198     \spacefactor\@savsf
199     \ifdim\@savsk>\z@
200       \nobreak \hskip\z@skip
201       \@ignoretrue
202       \ignorespaces
203     \fi
204   \fi}%
205 〈/2ekernel | \latexrelease〉
206 〈\latexrelease〉\EndIncludeInRelease
207 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
208 〈\latexrelease〉          {\@Eshack}{hyphenation after space hack}%
209 〈\latexrelease〉\def\@Eshack{%
210 〈\latexrelease〉  \relax
211 〈\latexrelease〉  \ifhmode

```

```

212 〈\latexrelease〉      \spacefactor\@savsf
213 〈\latexrelease〉      \ifdim\@savsk>\z@
214 〈\latexrelease〉          \ignorespacestrue
215 〈\latexrelease〉          \ignorespaces
216 〈\latexrelease〉          \fi
217 〈\latexrelease〉          \fi}%
218 〈\latexrelease〉\EndIncludeInRelease
219 {*2ekernel}

```

(End of definition for \@EspHack.)

\@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 of definition for \@vbsphack.)

1.5 Vertical spacing

L^AT_EX supports the plain T_EX commands \smallskip, \medskip and \bigskip. However, it redefines them using \vspace instead of \vskip.

Extra vertical space is added by the command \addvspace{\<skip>}, which adds a vertical skip of <skip> to the document. The sequence \addvspace{\<s1>} \addvspace{\<s2>} is equivalent to \addvspace{\<maximum of s1, s2>}.

\addvspace should be used only in vertical mode, and gives an error if it's not. The \addvspace command does *not* add vertical space if @minipage is true. The minipage environment uses this to inhibit the addition of extra vertical space at the beginning.

Penalties are put into the vertical list with the \addpenalty{\<penalty>} command. It works properly when \addpenalty and \addvspace commands are mixed.

The @nobreak switch is set true used when in vertical mode and no page break should occur. (Right now, it is used only by the section heading commands to inhibit page breaking after a heading.)

\@xaddvskip Internal macro for \vspace handling the case that space has previously been added.

```

220 \def\@xaddvskip{%
221   \ifdim\lastskip<\@tempskip
222     \vskip-\lastskip

```

```

223      \vskip\@tempskipb
224  \else
225      \ifdim\@tempskipb<\z@
226          \ifdim\lastskip<\z@
227              \else
228                  \advance\@tempskipb\lastskip
229                  \vskip-\lastskip
230                  \vskip \@tempskipb
231              \fi
232          \fi
233      \fi}

```

(End of definition for `\@xaddvskip`.)

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

```

234  </2ekernel>
235  <*2ekernel | latexrelease>
236  <latexrelease> \IncludeInRelease{2024/11/01}%
237  <latexrelease>           {\addvspace}{drop unnecessary no-item error}%
238  \protected\def\addvspace#1{%

```

When this is encountered in hmode, we check whether we are in an hbox and if so generate a L^AT_EX error, as otherwise this would cause a bunch of low-level errors. In unrestricted hmode we simply switch to vmode by issuing a `\par`.

```

239  \ifhmode \ifinner \@LRmoderr \else \par \fi \fi
240  \if@minipage\else
241      \ifdim \lastskip =\z@
242          \vspace@\calcify{\#1}%
243      \else
244          \setlength\@tempskipb{\#1}%
245          \vskip\@xaddvskip
246      \fi
247  \fi
248 }
249 </2ekernel | latexrelease>
250 <latexrelease> \EndIncludeInRelease
251 <latexrelease> \IncludeInRelease{2020/10/01}%
252 <latexrelease>           {\addvspace}{\addvspace calc support}%
253 <latexrelease> \def\addvspace#1{%
254 <latexrelease> \ifvmode
255 <latexrelease>     \if@minipage\else
256 <latexrelease>         \ifdim \lastskip =\z@
257 <latexrelease>             \vspace@\calcify{\#1}%
258 <latexrelease>         \else
259 <latexrelease>             \setlength\@tempskipb{\#1}%
260 <latexrelease>             \vskip\@xaddvskip
261 <latexrelease>         \fi
262 <latexrelease>     \fi
263 <latexrelease> \else
264 <latexrelease>     \noitemerr
265 <latexrelease> \fi}
266 <latexrelease> \EndIncludeInRelease

```

```

267 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
268 〈\latexrelease〉                                {\addvspace}{\addvspace calc support}%
269 〈\latexrelease〉
270 〈\latexrelease〉\def\addvspace#1{%
271 〈\latexrelease〉  \ifvmode
272 〈\latexrelease〉    \if@minipage\else
273 〈\latexrelease〉      \ifdim \lastskip =\z@
274 〈\latexrelease〉        \vskip #1\relax
275 〈\latexrelease〉      \else
276 〈\latexrelease〉        \tempskipb#1\relax
277 〈\latexrelease〉        \xaddvskip
278 〈\latexrelease〉      \fi
279 〈\latexrelease〉    \fi
280 〈\latexrelease〉  \else
281 〈\latexrelease〉    \noitemerr
282 〈\latexrelease〉  \fi}
283 〈\latexrelease〉\EndIncludeInRelease
284 〈*2ekernel〉

```

(End of definition for \addvspace.)

\addpenalty

```

285 〈/2ekernel〉
286 〈\latexrelease〉\IncludeInRelease{2024/11/01}%
287 〈\latexrelease〉                                {\addpenalty}{\addpenalty drop error}%
288 〈*2ekernel | \latexrelease〉
289 \protected\def\addpenalty#1{%

```

See description of \addvspace for documentation of the next line of code.

```
290 \ifhmode \ifinner \@LRmoderr \else \par \fi \fi
```

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.

```

291 \if@minipage
292 \else
293   \if@nobreak
294   \else
295     \ifdim\lastskip=\z@
296       \penalty#1\relax
297     \else
298       \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; that should not affect the skip we pass on to TeX.

```

299 \begingroup
300 \tempskipa\tempskipb
301 \advance \tempskipb
302 \ifdim\prevdepth>\maxdepth\maxdepth\else

```

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

```
303           \ifdim \prevdepth = -\z@ \else \prevdepth \fi
304           \fi
305           \vskip -\tempskipb
306           \penalty#1%
307           \ifdim\tempskipa=\tempskipb
```

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

```
308           \else
```

Combine the prevdepth adjustment into a single skip.

```
309           \advance\tempskipb -\tempskipa
310           \vskip \tempskipb
311           \fi
```

The final skip is always the specified length.

```
312           \vskip \tempskipa
313           \endgroup
314           \fi
315           \fi
316           \fi
317       }
318   {/2ekernel | latexrelease}
319   \end{IncludeInRelease}
320   \IncludeInRelease{2015/01/01}%
321   \def\addpenalty{\addpenalty}%
322   \def\def\addpenalty{\addpenalty}%
323   \ifvmode
324   \minipage
325   \else
326   \nobreak
327   \else
328   \ifdim\lastskip=\z@
329   \penalty#1\relax
330   \else
331   \tempskipb\lastskip
332   \begingroup
333   \tempskipa\tempskipb
334   \advance \tempskipb
335   \ifdim\prevdepth>\maxdepth\maxdepth\else
336   \ifdim \prevdepth = -\z@ \else \prevdepth \fi
337   \fi
338   \vskip -\tempskipb
339   \penalty#1%
340   \ifdim\tempskipa=\tempskipb
341   \else
342   \advance\tempskipb -\tempskipa
343   \vskip \tempskipb
344   \fi
345   \vskip \tempskipa
346   \endgroup
347   \fi
348   \fi
349   \fi
350   \else
```

```

351 〈\latexrelease〉    \noitemerr
352 〈\latexrelease〉  \fi}%
353 〈\latexrelease〉\EndIncludeInRelease
354 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
355 〈\latexrelease〉          {\addpenalty}{\addpenalty}%
356 〈\latexrelease〉\def\addpenalty#1{%
357 〈\latexrelease〉  \ifvmode
358 〈\latexrelease〉    \if@minipage
359 〈\latexrelease〉    \else
360 〈\latexrelease〉      \if@nobreak
361 〈\latexrelease〉      \else
362 〈\latexrelease〉        \ifdim\lastskip=\z@%
363 〈\latexrelease〉          \penalty#1\relax
364 〈\latexrelease〉        \else
365 〈\latexrelease〉          \tempskipb\lastskip
366 〈\latexrelease〉          \vskip -\lastskip
367 〈\latexrelease〉          \penalty#1%
368 〈\latexrelease〉          \vskip\tempskipb
369 〈\latexrelease〉        \fi
370 〈\latexrelease〉      \fi
371 〈\latexrelease〉    \fi
372 〈\latexrelease〉  \else
373 〈\latexrelease〉    \noitemerr
374 〈\latexrelease〉  \fi}%
375 〈\latexrelease〉\EndIncludeInRelease
376 〈*/2ekernel〉

```

(End of definition for `\addpenalty`.)

`\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.

```

377 \DeclareRobustCommand\vspace{\@ifstar\@vspacer\@vspace}
378 〈/2ekernel〉
379 〈*2ekernel | \textrule〉
380 〈\textrule〉\IncludeInRelease{2020/10/01}%
381 〈\textrule〉          {\@vspace}{Support calc in \vspace}%

```

We support calc syntax in the argument and therefore use `\setlength`.

```

382 \def\@vspace #1{%
383   \ifvmode
384     \@vspace@calcify{#1}%
385     \vskip\z@skip
386   \else
387     \bphack
388     \vadjust{\restorepar
389       \@vspace@calcify{#1}%
390       \vskip\z@skip

```

```

391          }%
392          \@esphack
393          \fi}
394 \def\@vspacer#1{%
395   \ifvmode
396     \dimen@\prevdepth
397     \hrule \@height\z@
398     \nobreak
399     \@vspace@calcify{#1}%
400     \vskip\z@skip
401     \prevdepth\dimen@
402   \else
403     \@bsphack
404     \vadjust{\@restorepar
405       \hrule \@height\z@
406       \nobreak
407       \@vspace@calcify{#1}%
408       \vskip\z@skip}%
409     \@esphack
410   \fi}
411 </2ekernel | latexrelease>
412 <latexrelease>\EndIncludeInRelease
413 <latexrelease>\IncludeInRelease{0000/00/00}%
414 <latexrelease>           {\@vspace}{Support calc in \vspace}%
415 <latexrelease>
416 <latexrelease>\def\@vspace #1{%
417 <latexrelease>  \ifvmode
418 <latexrelease>    \vskip #1
419 <latexrelease>    \vskip\z@skip
420 <latexrelease>  \else
421 <latexrelease>    \@bsphack
422 <latexrelease>    \vadjust{\@restorepar
423 <latexrelease>      \vskip #1
424 <latexrelease>      \vskip\z@skip
425 <latexrelease>    }%
426 <latexrelease>    \@esphack
427 <latexrelease>  \fi}
428 <latexrelease>\def\@vspacer#1{%
429 <latexrelease>  \ifvmode
430 <latexrelease>    \dimen@\prevdepth
431 <latexrelease>    \hrule \@height\z@
432 <latexrelease>    \nobreak
433 <latexrelease>    \vskip #1
434 <latexrelease>    \vskip\z@skip
435 <latexrelease>    \prevdepth\dimen@
436 <latexrelease>  \else
437 <latexrelease>    \@bsphack
438 <latexrelease>    \vadjust{\@restorepar
439 <latexrelease>      \hrule \@height\z@
440 <latexrelease>      \nobreak
441 <latexrelease>      \vskip #1
442 <latexrelease>      \vskip\z@skip}%
443 <latexrelease>    \@esphack
444 <latexrelease>  \fi}

```

```

445  ⟨{latexrelease}⟩\EndIncludeInRelease
446  ⟨{*2ekernel}⟩

(End of definition for \vspace, \@vspace, and \@vspace@.)
```

```

\smallskip
\medskip 447 \def\smallskip{\vspace{\smallskipamount}}
\bigskip 448 \def\medskip{\vspace{\medskipamount}}
449 \def\bigskip{\vspace{\bigskipamount}}
```

(End of definition for \smallskip, \medskip, and \bigskip.)

```

\smallskipamount
\medskipamount 450 \newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
\bigskipamount 451 \newskip\medskipamount \medskipamount =6pt plus 2pt minus 2pt
452 \newskip\bigskipamount \bigskipamount =12pt plus 4pt minus 4pt
```

(End of definition for \smallskipamount, \medskipamount, and \bigskipamount.)

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 space-factor; so use it carefully!

```

453 \DeclareRobustCommand{\nobreakdashes}{%
454   \leavevmode
455   \toks@{}%
456   \def\reserved@a##1{\toks@\expandafter{\the\toks@-}%
457     \futurelet\@let@token \reserved@b}%
458   \def\reserved@b {\ifx\@let@token -%
459     \expandafter\reserved@a
460   \else
461     \setbox\z@\hbox{\the\toks@\nobreak}%
462     \unhbox\z@
463     \spacefactor\sffcode`-
464   \fi}%
465   \futurelet\@let@token \reserved@b
466 }
```

(End of definition for \nobreakdashes.)

\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 L^AT_EX internal commands.

The braces in the definition of `\nobreakspace` 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`.

The fragile version of `\nobreakspace` needs a brace group after `\nobreakspace` to prevent loss of spaces if it occurs in an expansion context. That's not an issue with the updated `\protected` definition, so we keep the code shorter and avoid that.

```

467 \DeclareRobustCommand{\nobreakspace}{%
468   \leavevmode\nobreak\ }
469 \catcode `~=13
470 </2ekernel>
471 <latexrelease>\IncludeInRelease{2023/11/01}%
472 <latexrelease>          {\tilde}{Protected tilde}%
473 <*2ekernel | latexrelease>
474 \protected\edef~{%
475   \noexpand\ifincname\noexpand\expandafter\string~%
476   \noexpand\else
477     \noexpand\expandafter\noexpand\nobreakspace
478   \noexpand\fi
479 }
480 </2ekernel | latexrelease>
481 <latexrelease>\EndIncludeInRelease
482 <latexrelease>\IncludeInRelease{0000/00/00}%
483 <latexrelease>          {\tilde}{Protected tilde}%
484 <latexrelease>\def~{\nobreakspace{}}
485 <latexrelease>\EndIncludeInRelease
486 <*2ekernel>
487 \expandafter\let\expandafter\@xobeysp\csname nobreakspace \endcsname

```

(End of definition for `\nobreakspace` and `\@xobeysp`.)

`\@xobeystab` Equivalent to the space case with the default settings.

```

488 </2ekernel>
489 <latexrelease>\IncludeInRelease{2023/11/01}%
490 <latexrelease>          {\@xobeystab}{Obeyed tabs}%
491 <*2ekernel | latexrelease>
492 \let\@xobeystab\@xobeysp
493 </2ekernel | latexrelease>
494 <latexrelease>\EndIncludeInRelease
495 <latexrelease>\IncludeInRelease{0000/00/00}%
496 <latexrelease>          {\@xobeystab}{Obeyed tabs}%
497 <latexrelease>\let\@xobeystab\@undefined
498 <latexrelease>\EndIncludeInRelease
499 <*2ekernel>

```

(End of definition for `\@xobeystab`.)

`\@` 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.

```

500 </2ekernel>
501 <latexrelease>\IncludeInRelease{2015/01/01}%
502 <latexrelease>          {\@}{Space after \@}%
503 <*2ekernel | latexrelease>

```

```

504 \def\@{\spacefactor\@m{}}
505 </2ekernel | latexrelease>
506 <latexrelease>\EndIncludeInRelease
507 <latexrelease>\IncludeInRelease{0000/00/00}%
508 <latexrelease>           {\@}{Space after \@}%
509 <latexrelease>\def\@{\spacefactor\@m{}}
510 <latexrelease>\EndIncludeInRelease
511 <*2ekernel>

```

(End of definition for \@.)

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

(End of definition for \hspace.)

```
\@hspace
```

```

513 </2ekernel>
514 <*2ekernel | latexrelease>
515 <latexrelease>\IncludeInRelease{2020/10/01}%
516 <latexrelease>           {\@hspace}{Support calc with \hspace}%

```

We use a private register to calculate the space (if `calc` is used). Previously we used a group but that results in `\everypar` etc. being executed inside the group if the `\hspace` starts a paragraph. This is a bug fix so we do not provide rollback to the incorrect intermediate version.

```

517 \newskip\sp@ce@skip
518 \def\@hspace#1{\setlength\sp@ce@skip{#1}\hskip\sp@ce@skip}
519 </2ekernel | latexrelease>
520 <latexrelease>\EndIncludeInRelease
521 <latexrelease>\IncludeInRelease{0000/00/00}%
522 <latexrelease>           {\@hspace}{Support calc with \hspace}%
523
524 <latexrelease>
525 <latexrelease>\def\@hspace#1{\hskip #1\relax}
526 <latexrelease>\EndIncludeInRelease
527 <*2ekernel>
```

(End of definition for \@hspace.)

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

```

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

(End of definition for \@hspacer.)

```
\fill
```

```

530 \newskip\fill
531 \fill = Opt plus 1fill
```

(End of definition for \fill.)

```
\stretch
```

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

```

(End of definition for \stretch.)  

533 </2ekernel>  

534 <*2ekernel | latexrelease>  

535 <latexrelease>\IncludeInRelease{2018/12/01}%  

536 <latexrelease> {\thinspace}{Start LR-mode}%

\enspace  

537 \DeclareRobustCommand\enspace{\leavevmode@ifvmode\kern.5em }

(End of definition for \enspace.)  

\nleavevmode@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).  

538 \protected\def\leavevmode@ifvmode{\ifvmode\expandafter\indent\fi}

(End of definition for \leavevmode@ifvmode.)  

539 </2ekernel | latexrelease>  

540 <latexrelease>\EndIncludeInRelease  

541 <latexrelease>\IncludeInRelease{0000/00/00}%  

542 <latexrelease> {\thinspace}{Start LR-mode}%
543 <latexrelease>\def\thinspace{\kern.16667em }  

544 <latexrelease>\def\negthinspace{\kern-.16667em }  

545 <latexrelease>\def\enspace{\kern.5em }  

546 <latexrelease>\let\leavevmode@ifvmode@undefined  

547 <latexrelease>\EndIncludeInRelease  

548 <*2ekernel>

\enskip  

549 \def\enskip{\hskip.5em\relax}  

\quad 550 \def\quad{\hskip1em\relax}  

\quad 551 \def\quad{\hskip2em\relax}

(End of definition for \enskip, \quad, and \quad.)  

For Unicode engines, make the Unicode soft hyphen an active character defined as  

\-.  

552 \ifx\Umathcode\undefined\else  

553 \catcode "AD=13  

554 \def^^ad{\-}  

555 \fi

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

\restorecr  

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

(End of definition for \obeycr and \restorecr.)  

561 </2ekernel>

```

File 19

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.

```
1  <*2ekernel>
2  \DeclareRobustCommand{\TeX}{\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
```

(End of definition for \TeX.)

- \LaTeX The \LaTeX logo.

```
3  \DeclareRobustCommand{\LaTeX}{\kern-.36em%
4    {\sbox{z@T}%
5     \vbox to\ht{z@}{\hbox{\check@mathfonts%
6       \fontsize\sf@size\z@%
7       \math@fontsfalse\selectfont%
8       A}%
9       \vss}%
10  }%
11  \kern-.15em%
12  \TeX}
```

(End of definition for \LaTeX.)

- \LaTeXe The \LaTeX_{2ε} logo as proposed by A-W designers.

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

(End of definition for \LaTeXe.)

File 20

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 L ^A T _E X 2 _E , 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 <i>then</i> otherwise execute <i>else</i> . |
| \InputIfFileExists | \{(NAME)\}\{(then)\}\{(else)\} |
| | If the file exists on the system, execute <i>then</i> and input <i>NAME</i> otherwise execute <i>else</i> . |

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

1 <*2ekernel>
2 \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 \@files w = 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 \@files w = T
          then \immediate\openout\@partaux{FILE.AUX}
              \immediate\write\@partaux{\relax}
        fi
        \@input{FILE.TEX}
        \clearpage
        \@writeckpt{FILE}
        if @files w then \closeout\@partaux fi
        \@auxout := \@mainaux
    else \cp@FILE
  fi
END

\@writeckpt{FILE} ==
BEGIN
  if \@files w = 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 L^AT_EX 2.09 comments.

```
\@mainaux  
\@partaux  
 3 \newwrite\@mainaux  
 4 \newwrite\@partaux
```

(End of definition for \@mainaux and \@partaux.)

```
\if@filesw  
\if@partsw  
 5 \newif\if@filesw \@fileswtrue  
 6 \newif\if@partsw \@partswfalse
```

(End of definition for \if@filesw and \if@partsw.)

\@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).

```
 7 \newcount\@clubpenalty  
 8 \@clubpenalty \clubpenalty
```

(End of definition for \@clubpenalty.)

```
\document  
 9 </2ekernel>  
10 <latexrelease>\IncludeInRelease{2020/10/01}%  
11 <latexrelease> {\document}{Added hook to load l3backend code}%  
12 <2ekernel | latexrelease>  
13 \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.

```
14 % \endgroup  
15 \UseOneTimeHook{begindocument/before}-%  
16 \@kernel@after@begindocument@before
```

Added hook to load l3backend code:

```
17 \ExplSyntaxOn  
18 \ifx\@unusedoptionlist\empty\nothing  
19   \WarningNoLine{Unused global option(s): ^J}%  
20   \spaces[\@unusedoptionlist]}%  
21 \fi  
22 \colht\textheight  
23 \colroom\textheight \vsize\textheight  
24 \columnwidth\textwidth  
25 \clubpenalty\clubpenalty  
26 \if@twocolumn  
27   \advance\columnwidth -\columnsep
```

```

28      \divide\columnwidth\tw@ \hsize\columnwidth \iffirstcolumntrue
29  \fi
30  \hsize\columnwidth \linewidth\hsize
31  \begingroup\@floatplacement\@dblfloatplacement
32    \makeatletter\let\@writefile\gobbletwo
33    \global \let \multiplelabels \relax
34    \cinput{\jobname.aux}%
35  \endgroup
36  \if@files
37    \immediate\openout\mainaux\jobname.aux
38    \immediate\write\mainaux{\relax}%
39  \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.

```

40  \process@table
41  \let\glb@currsize\empty % Force math initialization.
42
43  \normalsize
44  \everypar{}%

```

So that punctuation in headings is not disturbed by verbatim or other local changes to the space factor codes, save the document default here. This will be locally reset by the output routine. For special cases a class may want to define `\normalsfcodes` directly, in case that definition will be used. (This is an old bug, problem existed in L^AT_EX2.0x and plain T_EX.)

```

44  \ifx\normalsfcodes\empty
45  \ifnum\sfcodes`_=\@m
46    \let\normalsfcodes\frenchspacing
47  \else
48    \let\normalsfcodes\nonfrenchspacing
49  \fi
50  \fi

```

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

```

51  \ifx\document@default@language\m@ne
52    \chardef\document@default@language\language
53  \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.

```

54  \@noskipsecfalse
55  \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...

```

56  \@kernel@before@begindocument
57  \UseOneTimeHook{begindocument}%
58  \@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.

```

59  \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
60  \global\@maxdepth\maxdepth
61  \global\let\@begindocumenthook\@undefined
62  \ifx\@listfiles\@undefined
63    \global\let\@filelist\relax
64    \global\let\@addtofilelist\@gobble
65  \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.

```

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

```

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

```

68  \global\let \@nодокумент \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.

```

69  \global\let\do\noexpand
70  \UseOneTimeHook{begindocument/end}%

```

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

```

71  \ignorespaces

```

Provide a global definition for `\do` as well, so that it is already defined in the preamble and not late as `\begin{document}` overwriting some definition given by the unsuspecting user in the preamble.

```

72  \let\do\noexpand

```

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`.

```

73  \NewHook{begindocument}
74  \NewHook{begindocument/before}
75  \NewHook{begindocument/end}

```

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 coding is somewhat obscure due to the name of the macro which requires constructing.

```

76 \edef \@kernel@after@begindocument@before {%
77   \let\expandafter\noexpand\csname
78     __hook env/document/begin\endcsname
79   \noexpand\empty}
80 %\let \@kernel@before@begindocument \empty
81 %\let \@kernel@after@begindocument \empty
82 (./2ekernel | latexrelease)
83 (latexrelease)\EndIncludeInRelease
84 (latexrelease)\IncludeInRelease{2017/04/15}%
85 (latexrelease) {\document}{Save language for hyphenation}%
86 (latexrelease)
87 (latexrelease)\def\document{\endgroup
88 (latexrelease) \ifx\@unusedoptionlist\empty\else
89 (latexrelease)   \@latex@warning@no@line{Unused global option(s):`^J'}%
90 (latexrelease)   \@spaces[\@unusedoptionlist]}%
91 (latexrelease) \fi
92 (latexrelease) \colht\textheight
93 (latexrelease) \colroom\textheight \vsize\textheight
94 (latexrelease) \columnwidth\textwidth
95 (latexrelease) \clubpenalty\clubpenalty
96 (latexrelease) \if@twocolumn
97 (latexrelease)   \advance\columnwidth -\columnsep
98 (latexrelease)   \divide\columnwidth\tw@ \hsize\columnwidth \firstcolumntrue
99 (latexrelease) \fi
100 (latexrelease) \hsize\columnwidth \linewidth\hsize
101 (latexrelease) \begingroup\flo@placement\dblfloatplacement
102 (latexrelease)   \makeatletter\let\@writetwo\gobbletwo
103 (latexrelease)   \global\let\@multiplelabels\relax
104 (latexrelease)   \input{\jobname.aux}%
105 (latexrelease) \endgroup
106 (latexrelease) \if@filesw
107 (latexrelease)   \immediate\openout\mainaux\jobname.aux
108 (latexrelease)   \immediate\write\mainaux{\relax}%
109 (latexrelease) \fi
110 (latexrelease) \process@table
111 (latexrelease) \let\glb@currsize\empty % Force math initialization.
112 (latexrelease) \normalsize
113 (latexrelease) \everypar{}%
114 (latexrelease) \ifx\normalsfcodes\empty
115 (latexrelease)   \ifnum\sfcodes`.=\@m
116 (latexrelease)     \let\normalsfcodes\frenchspacing
117 (latexrelease)   \else
118 (latexrelease)     \let\normalsfcodes\nonfrenchspacing
119 (latexrelease)   \fi
120 (latexrelease) \fi
121 (latexrelease) \ifx\document@default@language\m@ne
122 (latexrelease)   \chardef\document@default@language\language
123 (latexrelease) \fi

```

```

124 〈latexrelease〉 \c@noskipsecfalse
125 〈latexrelease〉 \let \orefundefined \relax
126 〈latexrelease〉 \let\AtBeginDocument\@firstofone
127 〈latexrelease〉 \begindocumenthook
128 〈latexrelease〉 \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
129 〈latexrelease〉 \global\c@maxdepth\maxdepth
130 〈latexrelease〉 \global\let\@begindocumenthook\@undefined
131 〈latexrelease〉 \ifx\@listfiles\@undefined
132 〈latexrelease〉 \global\let\@filelist\relax
133 〈latexrelease〉 \global\let\@addtofilelist\@gobble
134 〈latexrelease〉 \fi
135 〈latexrelease〉 \gdef\do##1{\global\let ##1\@notprerr}%
136 〈latexrelease〉 \c@preamblecmds
137 〈latexrelease〉 \global\let \c@nodocument \relax
138 〈latexrelease〉 \global\let\do\noexpand
139 〈latexrelease〉 \ignorespaces}
140 〈latexrelease〉 \EndIncludeInRelease
141 〈latexrelease〉
142 〈latexrelease〉 \IncludeInRelease{0000/00/00}%
143 〈latexrelease〉 {\document}{Save language for hyphenation}
144 〈latexrelease〉 \def\document{\endgroup
145 〈latexrelease〉 \ifx\@unusedoptionlist\empty\else
146 〈latexrelease〉 \c@latex@warning@no@line{Unused global option(s):`^J`%
147 〈latexrelease〉 \c@spaces[\@unusedoptionlist]}%
148 〈latexrelease〉 \fi
149 〈latexrelease〉 \c@colht\textheight
150 〈latexrelease〉 \c@colroom\textheight \vsize\textheight
151 〈latexrelease〉 \columnwidth\textwidth
152 〈latexrelease〉 \c@clubpenalty\clubpenalty
153 〈latexrelease〉 \if@twocolumn
154 〈latexrelease〉 \advance\columnwidth -\columnsep
155 〈latexrelease〉 \divide\columnwidth\tw@ \hsize\columnwidth
156 〈latexrelease〉 \c@firstcolumntrue
157 〈latexrelease〉 \fi
158 〈latexrelease〉 \hsize\columnwidth \linewidth\hsize
159 〈latexrelease〉 \begingroup\c@floatplacement\c@dblfloatplacement
160 〈latexrelease〉 \makeatletter\let\c@writefile\c@gobbletwo
161 〈latexrelease〉 \global\let\c@multipletables\relax
162 〈latexrelease〉 \c@input{\jobname.aux}%
163 〈latexrelease〉 \endgroup
164 〈latexrelease〉 \if@filesw
165 〈latexrelease〉 \immediate\openout\c@mainaux\jobname.aux
166 〈latexrelease〉 \immediate\write\c@mainaux{\relax}%
167 〈latexrelease〉 \fi
168 〈latexrelease〉 \process@table
169 〈latexrelease〉 \let\glb@currsize\empty
170 〈latexrelease〉 \normalsize
171 〈latexrelease〉 \everypar{}%
172 〈latexrelease〉 \ifx\normalsfcodes\empty
173 〈latexrelease〉 \c@ifnum\sfcode`\.=\c@m
174 〈latexrelease〉 \c@let\normalsfcodes\frenchspacing
175 〈latexrelease〉 \c@else
176 〈latexrelease〉 \c@let\normalsfcodes\nonfrenchspacing
177 〈latexrelease〉 \fi

```

```

178  \if{latexrelease}  \fi
179  \if{latexrelease}  \c@noskipsecfalse
180  \if{latexrelease}  \let \c@refundefined \relax
181  \if{latexrelease}  \let \AtBeginDocument \c@firstofone
182  \if{latexrelease}  \c@begindocumenthook
183  \if{latexrelease}  \ifdim \topskip < 1sp \global \topskip 1sp \relax \fi
184  \if{latexrelease}  \global \c@maxdepth \maxdepth
185  \if{latexrelease}  \global \let \c@begindocumenthook \c@undefined
186  \if{latexrelease}  \ifx \c@listfiles \c@undefined
187  \if{latexrelease}    \global \let \c@filelist \relax
188  \if{latexrelease}    \global \let \c@addtolist \c@gobble
189  \if{latexrelease}  \fi
190  \if{latexrelease}  \gdef \do{\#1} {\global \let \do \c@notprerr}%
191  \if{latexrelease}  \c@preamblecmds
192  \if{latexrelease}  \global \let \c@nodocument \relax
193  \if{latexrelease}  \global \let \do \c@noexpand
194  \if{latexrelease}  \ignorespaces}
195  \if{latexrelease} \EndIncludeInRelease
196  {*2ekernel}
197  \c@onlypreamble \document

```

(End of definition for *\document* and others.)

\normalsfcodes The setting of *\c@empty* is just a flag. This command may be defined in a class or package file. If it is still *\c@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.

```
198 \let \normalsfcodes \c@empty
```

(End of definition for *\normalsfcodes*.)

\nofiles Set *\c@fileswfalse* which suppresses the places where L^AT_EX makes *\immediate* writes. The *\makeindex* and *\maketoc* 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@cnobreak* test is needed.

```

199 \def \nofiles{%
200   \c@fileswfalse
201   \typeout{No auxiliary output files.^J}%
202   \long\def \protected@write{\#1##2##3{%
203     {\c@write \m@ne \c@if nobreak \c@if vmode \c@nobreak \c@fi \c@fi}%
204   \let \makeindex \relax
205   \let \maketoc \relax
206 } \c@onlypreamble \nofiles

```

(End of definition for *\nofiles*.)

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

```

207 \long\def \protected@write{\#1##2##3{%
208   \begingroup
209     \let \thepage \relax
210     #2%
211     \let \protect \c@unexpandable \protect
212     \edef \reserved@a {\c@write \#1 \c@#3}%

```

```

213      \reserved@a
214      \endgroup
215      \if@nobreak\ifvmode\nobreak\fi\fi
216 }

```

(End of definition for `\protected@write`.)

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

`\include` In the definition of `\include`, `\def\reserved@b` changed to `\edef\reserved@b` to be
`\includeonly` consistent with the `\edef` in `\includeonly`. (Suggested by Rainer Schöpf & Frank
Mittelbach. Change made 20 Jul 88.)

Changed definition of `\include` to allow space at end of file name — otherwise,
typing `\include{foo }` would cause L^AT_EX to overwrite `foo.tex`. Change made 24 May
89, suggested by Rainer Schöpf and Frank Mittelbach

Made `\include` check for being used inside an `\include`'d file, as this will not work
and cause surprising results.

```

218 </2ekernel>
219 <*2ekernel | latexrelease>
220 <|latexrelease>\IncludeInRelease{2020/10/01}%
221 <|latexrelease>                                {\includeonly}{Spaces in file names}%
222 \def\include#1{\relax
223   \ifnum\@auxout=\@partaux
224     \@latex@error{\string\include\space cannot be nested}\@eha
225   \else

```

Here the normalization will add `.tex` for all files, (it uses the same normalization as the
hooks), so we need to remove that manually. `\@strip@tex@ext` does that.

```

226   \set@curr@file{#1}%
227   \edef\@curr@file{\@strip@tex@ext\@curr@file}%

```

For historical reasons `\@include` expects an argument delimited by a space. This is kept
(though unnecessary now) to avoid errors in other packages that use `\@include` directly.

```

228   \expandafter\@include\expandafter{\@curr@file} % deliberate space
229   \fi}

```

Here in `\includeonly` we also need to strip `.tex` after normalization:

```

230 \def\includeonly#1{%
231   \@partstrue

```

Because the argument to `\includeonly` is a comma-separated list of filenames where
there may be comma's preceding 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.

```

232   \let\@partlist\@empty
233   \@for\reserved@a:=#1 \do
234   {%
235     \expandafter\set@curr@file\expandafter{\reserved@a}%
236     \ifx\@partlist\@empty
237       \edef\@partlist{\@strip@tex@ext\@curr@file}%
238     \else
239       \edef\@partlist{\@partlist,\@strip@tex@ext\@curr@file}%
240     \fi
241   }%
242 }
243 \@onlypreamble\includeonly

```

(End of definition for \include and \includeonly.)

\@strip@tex@ext These macros take a (\detokenized 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.

```
244 \def\reserved@a#1{%
245 \@strip@tex@ext##1{%
246 \expandafter\@strip@tex@ext@aux
247 ##1\@nil\@nil
248 #1\@nil\relax\@nnil}
249 \def\@strip@tex@aux##1#1\@nil##2\@nnil{%
250 \ifx\relax##2\@empty
251 \expandafter\@cdr\expandafter\@empty\@cdr{}##1%
252 \else##1\fi}%
253 \expandafter\reserved@a
254 \expandafter{\detokenize{.tex}}
255 
```

(End of definition for \@strip@tex@ext and \@strip@tex@ext@aux.)

```
256 \end{tex} \EndIncludeInRelease
257 \end{tex} \IncludeInRelease{2019/10/01}%
258 \end{tex} {\includeonly}{Spaces in file names}%
259 \end{tex}
260 \end{tex} \def\includeonly#1{%
261 \end{tex} \partswtrue
262 \end{tex} \set@curr@file{\zap@space#1 \empty}%
263 \end{tex} \let\@partlist\curr@file
264 \end{tex} }
265 \end{tex}
266 \end{tex} \def\include#1{\relax
267 \end{tex} \ifnum\auxout=\partaux
268 \end{tex} \@latex@error{\string\include\space cannot be nested}\eha
269 \end{tex} \else
270 \end{tex} \set@curr@file{#1 }%
271 \end{tex} \expandafter\@include\curr@file
272 \end{tex} \fi}
273 \end{tex}
274 \end{tex} \let\@strip@tex@ext\undefined
275 \end{tex} \let\@strip@tex@ext@aux\undefined
276 \end{tex}
277 \end{tex} \EndIncludeInRelease
278 \end{tex} \IncludeInRelease{0000/00/00}%
279 \end{tex} {\includeonly}{Spaces in file names}%
280 \end{tex} \def\includeonly#1{%
281 \end{tex} \partswtrue
282 \end{tex} \edef\@partlist{\zap@space#1 \empty}%
283 \end{tex}
284 \end{tex} \def\include#1{\relax
285 \end{tex} \ifnum\auxout=\partaux
286 \end{tex} \@latex@error{\string\include\space cannot be nested}\eha
287 \end{tex} \else \include#1 \fi}
288 \end{tex}
289 \end{tex} \EndIncludeInRelease
290 
```

```

\@include

291  </2ekernel>
292  {*2ekernel | latexrelease}
293  {latexrelease}\IncludeInRelease{2022/06/01}%
294  {latexrelease}          {\@include}{Spaces in file names and hooks}%

295 \def\@include#1 {%
296 \ifx\@nodocument\relax
297   \clearpage
298   \if@filesw
299     \immediate\write\@mainaux{\string\@input{#1.aux}}%
300   \fi
301   \tempswattrue
302   \if@partsw
303     \tempswafalse
304     \edef\reserved@b{#1}%
305     \for\reserved@a:=\partlist\do
306       {\ifx\reserved@a\reserved@b\tempswattrue\fi}%
307   \fi
308   \if@tempswa
309     \let\auxout\partaux
310     \if@filesw
311       \immediate\openout\partaux "#1.aux"
312       \immediate\write\partaux{\relax}%
313     \fi

```

Now before going to the hooks we need to set `\CurrentFile`:

```

314 %-----%
315 \filehook@set@CurrentFile

```

Execute the `before` hooks just after we switched the `.aux` file ...

```

316   \UseHook{include/before}%
317   \UseOneTimeHook{include/#1/before}%
318 %-----%
319   \input{#1.tex}%
320 %-----%
... then end hooks ...
321   \UseOneTimeHook{include/#1/end}%
322   \UseHook{include/end}%
323 %-----%
324   \clearpage
325 %-----%

```

... 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`).

```

326   \UseOneTimeHook{include/#1/after}%
327   \UseHook{include/after}%
328   \clearpage
329 %-----%
330   \writeckpt{#1}%
331   \if@filesw
332     \immediate\closeout\partaux

```

```

333     \fi
334 \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.

```

335     \deadcycles\z@
336     \curnameuse{cp@#1}\%
```

We also execute a hook in this case, first a general used for every include file that is exclude and then a specific one that contains the name of the include file.

```

337 %-----%
338     \UseHook{include/excluded}%
339     \UseOneTimeHook{include/#1/excluded}%
340 %-----%
341     \fi
342     \let\@auxout\@mainaux
343 \else
344 \@latex@warning{%
345   \noexpand\include should only be used after \string\begin{document}}%
346 \cinput@{#1}%
347 \fi}
```

Now declare the non-generic `include` hooks used above:

```

348 \NewHook{include/before}
349 \NewReversedHook{include/end}
350 \NewReversedHook{include/after}
351 \NewHook{include/excluded}
352 <latexrelease> \EndIncludeInRelease
353 (/2ekernel | latexrelease)

354 <latexrelease> \IncludeInRelease{2020/10/01}%
355 <latexrelease>           {:@include}{Spaces in file names and hooks}%
356 <latexrelease> \EndIncludeInRelease
357 <latexrelease> \def\@include#1 {%
358   <latexrelease> \ifx\@nodocument\relax
359   <latexrelease>   \clearpage
360   <latexrelease>   \if@filesw
361   <latexrelease>     \immediate\write\@mainaux{\string\@input{#1.aux}}%
362   <latexrelease>   \fi
363   <latexrelease>   \tempswattrue
364   <latexrelease>   \if@partsw
365   <latexrelease>     \tempswafalse
366   <latexrelease>     \edef\reserved@b{#1}%
367   <latexrelease>     \for\reserved@a:=\partlist\do
368   <latexrelease>       {\ifx\reserved@a\reserved@b\tempswattrue\fi}%
369   <latexrelease>   \fi
370   <latexrelease>   \if@tempswa
371   <latexrelease>     \let\@auxout\partaux
372   <latexrelease>     \if@filesw
373   <latexrelease>       \immediate\openout\partaux "#1.aux"
374   <latexrelease>       \immediate\write\partaux{\relax}%
375   <latexrelease>     \fi
376   <latexrelease>     \filehook@set@CurrentFile
377   <latexrelease>     \UseHook{include/before}%
378   <latexrelease>     \UseOneTimeHook{include/#1/before}%

```

```

379 <{latexrelease}>     \@input@{#1.tex}%
380 <{latexrelease}>     \UseOneTimeHook{include/#1/end}%
381 <{latexrelease}>     \UseHook{include/end}%
382 <{latexrelease}>     \clearpage
383 <{latexrelease}>     \UseOneTimeHook{include/#1/after}%
384 <{latexrelease}>     \UseHook{include/after}%
385 <{latexrelease}>     \clearpage
386 <{latexrelease}>     \@writeckpt{#1}%
387 <{latexrelease}>     \if@filesw
388 <{latexrelease}>         \immediate\closeout\@partaux
389 <{latexrelease}>     \fi
390 <{latexrelease}> \else
391 <{latexrelease}>     \deadcycles\z@
392 <{latexrelease}>     \@nameuse{cp@#1}%
393 <{latexrelease}> \fi
394 <{latexrelease}> \let\@auxout\@mainaux
395 <{latexrelease}\else
396 <{latexrelease}\@latex@warning{%
397 <{latexrelease}\ noexpand\include should only be used after \string\begin{document}}%
398 <{latexrelease}\@input@{#1}%
399 <{latexrelease}\fi}
400 <{latexrelease}\NewHook{include/before}%
401 <{latexrelease}\NewReversedHook{include/end}%
402 <{latexrelease}\NewReversedHook{include/after}%
403 <{latexrelease}\IncludeInRelease{0000/00/00}%
404 <{latexrelease}          {\@include}{Spaces in file names and hooks}%
405 <{latexrelease}\def\@include#1 {%
406 <{latexrelease}\ \clearpage
407 <{latexrelease}\ \if@filesw
408 <{latexrelease}\ \immediate\write\@mainaux{\string\@input{#1.aux}}%
409 <{latexrelease}\ \fi
410 <{latexrelease}\ \tempswattrue
411 <{latexrelease}\ \if@partsw
412 <{latexrelease}\ \tempswafalse
413 <{latexrelease}\ \edef\reserved@b{#1}%
414 <{latexrelease}\ \for\reserved@a:=\@partlist\do
415 <{latexrelease}\ \ifx\reserved@a\reserved@b\@tempswattrue\fi}%
416 <{latexrelease}\ \fi
417 <{latexrelease}\ \if@tempswa
418 <{latexrelease}\ \let\@auxout\@partaux
419 <{latexrelease}\ \if@filesw
420 <{latexrelease}\ \immediate\openout\@partaux #1.aux
421 <{latexrelease}\ \immediate\write\@partaux{\relax}%
422 <{latexrelease}\ \fi
423 <{latexrelease}\ \@input@{#1.tex}%
424 <{latexrelease}\ \clearpage
425 <{latexrelease}\ \@writeckpt{#1}%
426 <{latexrelease}\ \if@filesw
427 <{latexrelease}\ \immediate\closeout\@partaux
428 <{latexrelease}\ \fi
429 <{latexrelease}\ \else
430 <{latexrelease}\ \deadcycles\z@
431 <{latexrelease}\ \@nameuse{cp@#1}%
432 <{latexrelease}\ \fi

```

```

433 〈latexrelease〉 \let\@auxout\@mainaux}
434 〈latexrelease〉
435 〈latexrelease〉\EndIncludeInRelease
436 〈*2ekernel〉

```

(End of definition for \@include.)

\@writeckpt

```

437 \def\@writeckpt#1{%
438   \if@filesw
439     \immediate\write\@partaux{\string\@setckpt{#1}\@charlb}%
440     {\let\@elt\@wckptelt \cl@0@ckpt}%
441     \immediate\write\@partaux{\@charrb}%
442   \fi}

```

(End of definition for \@writeckpt.)

\@wckptelt

```

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

```

(End of definition for \@wckptelt.)

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

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

(End of definition for \@setckpt.)

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

```

447 {\catcode`[=1 \catcode`=2
448 \catcode`{=11 \catcode`}=11
449 \gdef\@charlb[{]
450 \gdef\@charrb[]}
451 }% }brace matching

```

(End of definition for \@charlb and \@charrb.)

1.1 Safe Input Macros

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

The definition below is from 2019 and only used during kernel bootstrapping, later on in `ltfilehook.dtx` it will get overwritten.

```

452 \def\set@curr@file#1{%
453   \begingroup
454     \escapechar\m@ne
455     \xdef\@curr@file{%
456       \expandafter\expandafter\expandafter\unquote@name
457       \expandafter\expandafter\expandafter{%
458         \expandafter\string
459           \csname@\firstofone#1\empty\endcsname}%
460     \endgroup
461   }

```

(End of definition for `\@curr@file` and `\set@curr@file`.)

`\quote@name` Quoting spaces

```

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

```

```

462 </2ekernel>
463 <*2ekernel | latexrelease>
464 <latexrelease>\IncludeInRelease{2019/10/01}%
465 <latexrelease>          {\quote@name}{Quote file names}%
466 \def\quote@name#1{"\quote@@name#1\@gobble"}%
467 \def\quote@@name#1"#1\quote@@name}

```

and removing quotes ...

```
468 \def\unquote@name#1{\quote@@name#1\@gobble}
```

(End of definition for `\quote@name`, `\quote@@name`, and `\unquote@name`.)

`\IfFileExists`

```

469 \DeclareRobustCommand\IfFileExists[1]{%
470   \set@curr@file{#1}%
471   \expandafter\IfFileExists@\expandafter{\@curr@file}}

```

(End of definition for `\IfFileExists`.)

```

472 </2ekernel | latexrelease>
473 <latexrelease>\EndIncludeInRelease
474 <latexrelease>\IncludeInRelease{0000/00/00}%
475 <latexrelease>          {\quote@name}{Quote file names}%
476 <latexrelease>
477 <latexrelease>\let\quote@name\@undefined
478 <latexrelease>\let\quote@@name\@undefined
479 <latexrelease>\let\unquote@name\@undefined
480 <latexrelease>
481 <latexrelease>\long\def \IfFileExists#1#2#3{%
482 <latexrelease>  \openin\@inputcheck#1 %
483 <latexrelease>  \ifeof\@inputcheck
484 <latexrelease>    \ifx\input@path\@undefined
485 <latexrelease>      \def\reserved@a{#3}%
486 <latexrelease>    \else

```

```

487 〈\latexrelease〉      \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
488 〈\latexrelease〉      \fi
489 〈\latexrelease〉      \else
490 〈\latexrelease〉      \closein\@inputcheck
491 〈\latexrelease〉      \edef\@filef@und{#1 }%
492 〈\latexrelease〉      \def\reserved@a{#2}%
493 〈\latexrelease〉      \fi
494 〈\latexrelease〉      \reserved@a}
495 〈\latexrelease〉
496 〈\latexrelease〉\EndIncludeInRelease
497 (*2ekernel〉

```

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

The original definition picked up arguments #2 and #3 in a way that they couldn't contain unbalanced conditionals. A better implementation would have been not to pick up the arguments at all but instead use the usual \@firstoftwo and \@secondoftwo. However, that changes how # is interpreted and so we can't do that nowadays without invalidating a lot of code. Therefore the somewhat curious construction near the end.

Earlier versions used \openin here, but this led to two code paths, one in expl3 and one here. To avoid that, and as the expl3 approach works by expansion, we use that here. As we need the file name to include the path, the actual expl3 function used is not the file existence test!

```

498 〈/2ekernel〉
499 〈*2ekernel | \latexrelease〉
500 〈\latexrelease〉\IncludeInRelease{2023/06/01}%
501 〈\latexrelease〉      {\@IfFileExists@}{\IfFileExists}%
502 \long\def \IfFileExists@#1#2#3{%
503   \edef\@filef@und{\IfFileExists@#1}%

```

The expl3 function regards an empty argument as nothing at all, but the L^AT_EX 2 _{ε} convention is that this is equal to the special .tex file.

```

504   \ifx\@filef@und\@empty
505     \if\relax\detokenize{#1}\relax
506       \let\reserved@a\@firstoftwo
507       \def\@filef@und{"tex"}%
508     \else
509       \let\reserved@a\@secondoftwo
510     \fi
511   \else
512     \let\reserved@a\@firstoftwo
513     \edef\@filef@und"\@filef@und" %
514   \fi

```

This is just there so that any # inside #2 or #3 needs doubling (as that was the case in the past).

```

515   \expandafter\def\expandafter\reserved@a
516     \expandafter{\reserved@a{#2}{#3}}%
517 \reserved@a}

```

Pipes are not really files, but at the document level they are supported. To quickly trim of any leading spaces, we use a blank test and \use:n rather than \tl_trim_spaces:n for speed as we don't care about the end of the input.

```

518 \ExplSyntaxOn
519 \cs_new:Npn \IfFileExists@@ #1

```

```

520  {
521    \tl_if_blank:nF {#1}
522    {
523      \tl_if_head_eq_charcode:oNTF { \use:n #1 } |
524      {#1}
525      { \file_full_name:n {#1} }
526    }
527  }
528 \cs_generate_variant:Nn \tl_if_head_eq_charcode:nNTF { o }
529 \ExplSyntaxOff
530 ⟨/2ekernel | latexrelease⟩
531 ⟨latexrelease⟩\EndIncludeInRelease
532 ⟨latexrelease⟩\IncludeInRelease{2021/06/01}%
533 ⟨latexrelease⟩
534 ⟨latexrelease⟩
535 ⟨latexrelease⟩\long\def \IfFileExists@#1#2#3{%
536 ⟨latexrelease⟩ \openin\@inputcheck"#1" %
537 ⟨latexrelease⟩ \ifeof\@inputcheck
538 ⟨latexrelease⟩ \ifx\input@path\@undefined
539 ⟨latexrelease⟩ \let\reserved@a\@secondoftwo
540 ⟨latexrelease⟩ \else
541 ⟨latexrelease⟩ \def\reserved@a{\@iffilenonpath{#1}}%
542 ⟨latexrelease⟩ \fi
543 ⟨latexrelease⟩ \else
544 ⟨latexrelease⟩ \closein\@inputcheck
545 ⟨latexrelease⟩ \edef\@filef@und{"#1"}%
546 ⟨latexrelease⟩ \let\reserved@a\@firstoftwo
547 ⟨latexrelease⟩ \fi
548 ⟨latexrelease⟩ \expandafter\def\expandafter\reserved@a
549 ⟨latexrelease⟩ \expandafter{\reserved@a{#2}{#3}}%
550 ⟨latexrelease⟩\reserved@a
551 ⟨latexrelease⟩\let\IfFileExists@@\@undefined
552 ⟨latexrelease⟩\EndIncludeInRelease
553 ⟨latexrelease⟩
554 ⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
555 ⟨latexrelease⟩
556 ⟨latexrelease⟩
557 ⟨latexrelease⟩\long\def \IfFileExists@#1#2#3{%
558 ⟨latexrelease⟩ \openin\@inputcheck"#1" %
559 ⟨latexrelease⟩ \ifeof\@inputcheck
560 ⟨latexrelease⟩ \ifx\input@path\@undefined
561 ⟨latexrelease⟩ \def\reserved@a{#3}%
562 ⟨latexrelease⟩ \else
563 ⟨latexrelease⟩ \def\reserved@a{\@iffilenonpath{#1}{#2}{#3}}%
564 ⟨latexrelease⟩ \fi
565 ⟨latexrelease⟩ \else
566 ⟨latexrelease⟩ \closein\@inputcheck
567 ⟨latexrelease⟩ \edef\@filef@und{"#1"}%
568 ⟨latexrelease⟩ \def\reserved@a{#2}%
569 ⟨latexrelease⟩ \fi
570 ⟨latexrelease⟩ \reserved@a
571 ⟨latexrelease⟩\EndIncludeInRelease
572 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
573 ⟨latexrelease⟩

```

```

574 〈\latexrelease〉
575 〈\latexrelease〉\let\IfFileExists@{\@undefined
576 〈\latexrelease〉
577 〈\latexrelease〉
578 〈\latexrelease〉\EndIncludeInRelease
579 {<2ekernel〉

(End of definition for \IfFileExists@ and \IfFileExists@@.)
```

\@iffileonpath If the file is not found by \openin, and \input@path is defined, look in all the directories specified in \input@path.

```

580 {</2ekernel>
581 {<2ekernel | \latexrelease>
582 〈\latexrelease〉\IncludeInRelease{2019/10/01}%
583 〈\latexrelease〉{\@iffileonpath}{\Quote file names}
584 \long\def\@iffileonpath#1{%
585   \let\reserved@a\@secondoftwo
586   \expandafter\@tfor\expandafter\reserved@b\expandafter
587     :\expandafter=\input@path\do{%
588     \openin\@inputcheck\expandafter\quote@name\expandafter{\reserved@b#1} %
589     \ifeof\@inputcheck\else
590       \edef\@filef@und{\expandafter\quote@name\expandafter{\reserved@b#1} }%
591       \let\reserved@a\@firstoftwo%
592       \closein\@inputcheck
593       \@break@tfor
594     \fi}%
595   \reserved@a}
```

(End of definition for \@iffileonpath.)

```

596 {</2ekernel | \latexrelease>
597 〈\latexrelease〉\EndIncludeInRelease
598 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
599 〈\latexrelease〉{\@quote@name}{\Quote file names}
600 〈\latexrelease〉
601 〈\latexrelease〉\long\def\@iffileonpath#1{%
602   \let\reserved@a\@secondoftwo
603   \expandafter\@tfor\expandafter\reserved@b\expandafter
604     :\expandafter=\input@path\do{%
605       \openin\@inputcheck\reserved@b#1 %
606       \ifeof\@inputcheck\else
607         \edef\@filef@und{\reserved@b#1 }%
608         \let\reserved@a\@firstoftwo%
609         \closein\@inputcheck
610         \@break@tfor
611       \fi}%
612   \reserved@a}
613 〈\latexrelease〉
614 〈\latexrelease〉\EndIncludeInRelease
615 {<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**.

```

616 \DeclareRobustCommand \InputIfFileExists[2]{%
617   \IfFileExists{#1}{%
618     {%
619       \expandafter\@swaptwoargs\expandafter
620         {\@filef@und}{#2\@addtofilelist{#1}\@@input}}}

```

(End of definition for `\InputIfFileExists`.)

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

```

621 〈/2ekernel〉
622 〈*2ekernel | latexrelease〉
623 〈latexrelease〉\IncludeInRelease{2019/10/01}%
624 〈latexrelease〉 {\@swaptwoargs}{Don't lose the file name}%
625 \long\def\@swaptwoargs#1#2{#2#1}

626 〈/2ekernel | latexrelease〉
627 〈latexrelease〉\EndIncludeInRelease
628 〈latexrelease〉\IncludeInRelease{0000/00/00}%
629 〈latexrelease〉 {\@swaptwoargs}{Don't lose the file name}%
630 〈latexrelease〉\let\@swaptwoargs\@undefined
631 〈latexrelease〉\EndIncludeInRelease
632 〈*2ekernel〉

```

(End of definition for `\@swaptwoargs`.)

`\input` Input a file: if the argument is given in braces use safe input macros, otherwise use TeX's primitive `\input` command (which is called `\@@input` in L^AT_EX).

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

(End of definition for `\input`.)

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

Changes to `\@iinput`: adapt to the changes to `\@missingfileerror`.

```

634 〈/2ekernel〉
635 〈*2ekernel | latexrelease〉
636 〈latexrelease〉\IncludeInRelease{2020/10/01}%
637 〈latexrelease〉 {\@iinput}{Change in file error handling}%
638 \def\@iinput#1{%
639   \InputIfFileExists{#1}{}%
640   {\filename@parse\@curr@file
641    \edef\reserved@a{\noexpand\@missingfileerror
642      {\filename@area\filename@base}%
643      {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%

```

This line now just sets `\@missingfile@{part}`:

```
644 \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.

```

645 \edef\reserved@a{\noexpand\@iinput{%
646   \@missingfile@area\@missingfile@base.\@missingfile@ext}{}%
647 \reserved@a}%
648 〈/2ekernel | latexrelease〉
649 〈latexrelease〉\EndIncludeInRelease

```

```

650 〈latexrelease〉\IncludeInRelease{2019/10/01}%
651 〈latexrelease〉          {\@iinput}{Quote file names}%
652 〈latexrelease〉
653 〈latexrelease〉\def\@iinput#1{%
654 〈latexrelease〉  \InputIfFileExists{#1}{}%
655 〈latexrelease〉  {\filename@parse@\curr@file
656 〈latexrelease〉  \edef\reserved@a{\noexpand\@missingfileerror
657 〈latexrelease〉  {\filename@area\filename@base}%
658 〈latexrelease〉  {\ifx\filename@ext\relax tex\else\filename@ext\fi}{}%
659 〈latexrelease〉  \reserved@a}%
660 〈latexrelease〉\EndIncludeInRelease
661 〈latexrelease〉\IncludeInRelease{0000/00/00}%
662 〈latexrelease〉          {\@iinput}{Quote file names}%
663 〈latexrelease〉\def\@iinput#1{%
664 〈latexrelease〉  \InputIfFileExists{#1}{}%
665 〈latexrelease〉  {\filename@parse{#1}%
666 〈latexrelease〉  \edef\reserved@a{\noexpand\@missingfileerror
667 〈latexrelease〉  {\filename@area\filename@base}%
668 〈latexrelease〉  {\ifx\filename@ext\relax tex\else\filename@ext\fi}{}%
669 〈latexrelease〉  \reserved@a}%
670 〈latexrelease〉\EndIncludeInRelease
671 〈/2ekernel〉

```

(End of definition for `\@iinput`.)

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

We don’t want `.aux`, `.toc` files etc be listed by `\listfiles`. However, something like `.bb1` probably should be listed and thus should be implemented not by `\@input`.

```

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

```

(End of definition for `\@input`.)

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

```

674 \def\@input@#1{\InputIfFileExists{#1}{}{\typeout{No file #1.}}}

```

(End of definition for `\@input@`.)

`\@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.

```

675 〈/2ekernel〉
676 〈*2ekernel | latexrelease〉
677 〈latexrelease〉\IncludeInRelease{2020/10/01}%
678 〈latexrelease〉  {\@missingfileerror}{Do not load missing file immediately}%
679 〈gdef\@missingfileerror#1#2{%
680   \typeout{^^J! LaTeX Error: File '#1.#2' not found.^^J^^J%
681   Type X to quit or <RETURN> to proceed,^^J%
682   or enter new name. (Default extension: #2)^^J}%
683   \message{Enter file name: }%
684   {\endlinechar\m@ne

```

```

685      \global\read\m@ne to\@gtempa}%
686      \ifx\@gtempa\@empty

```

If the user answers with `<return>`, fallback to the .tex file (previously it did nothing).

```

687      \let\@missingfile@area\@empty
688      \let\@missingfile@base\@empty
689      \def\@missingfile@ext{tex}%
690      \else

```

Use `\batchmode\read-1` to `<t1>` to end the TeX run, same as `expl3` does (it was `\batchmode\@@end` before).

```

691      \def\reserved@b{\batchmode\read-1 to \reserved@a}%
692      \def\reserved@a{x}\ifx\reserved@a\@gtempa\reserved@b\fi
693      \def\reserved@a{X}\ifx\reserved@a\@gtempa\reserved@b\fi
694      \filename@parse\@gtempa
695      \edef\filename@ext{%
696          \ifx\filename@ext\relax#2\else\filename@ext\fi}%
697      \edef\reserved@a{%

```

Only check `\IfFileExists` (it was `\InputIfFileExists`).

```

698      \noexpand\IfFileExists
699          {\filename@area\filename@base.\filename@ext}%

```

If the file exists, define `\@missingfile@{part}`.

```

700      {\def\noexpand\@missingfile@area{\filename@area}%
701      \def\noexpand\@missingfile@base{\filename@base}%
702      \def\noexpand\@missingfile@ext {\filename@ext}}%
703      {\noexpand\@missingfileerror
704          {\filename@area\filename@base}{\filename@ext}}%
705      \reserved@a
706      \fi
707  }
708  (/2ekernel | latexrelease)
709  (latexrelease)\EndIncludeInRelease
710  (latexrelease)\IncludeInRelease{0000/00/00}%
711  (latexrelease)      {(\@missingfileerror){Do not load missing file immediately}}%
712  (latexrelease)
713  (latexrelease)\gdef\@missingfileerror#1#2{%
714  (latexrelease)      \typeout{^^J! LaTeX Error: File '#1.#2' not found.^^J^^J%
715  (latexrelease)      Type X to quit or <RETURN> to proceed,^^J%
716  (latexrelease)      or enter new name. (Default extension: #2)^J}%
717  (latexrelease)      \message{Enter file name: }%
718  (latexrelease)      {\endlinechar\m@ne
719  (latexrelease)      \global\read\m@ne to\@gtempa}%
720  (latexrelease)      \ifx\@gtempa\@empty
721  (latexrelease)      \else
722  (latexrelease)      \def\reserved@a{x}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
723  (latexrelease)      \def\reserved@a{X}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
724  (latexrelease)      \filename@parse\@gtempa
725  (latexrelease)      \edef\filename@ext{%
726  (latexrelease)          \ifx\filename@ext\relax#2\else\filename@ext\fi}%
727  (latexrelease)      \edef\reserved@a{%
728  (latexrelease)          \noexpand\InputIfFileExists
729  (latexrelease)              {\filename@area\filename@base.\filename@ext}}%
730  (latexrelease)          {}}%

```

```

731 〈\latexrelease〉      {\noexpand\@missingfileerror
732 〈\latexrelease〉          {\filename@area\filename@base}\{\filename@ext\}}}}%
733 〈\latexrelease〉      \reserved@a
734 〈\latexrelease〉      \fi}
735 〈\latexrelease〉
736 〈\latexrelease〉\EndIncludeInRelease
737 {*2ekernel}

```

(End of definition for \@missingfileerror.)

\@obsoletefile For compatibility with L^AT_EX 2.09 document styles, we distribute files called `article.sty`, `book.sty`, `report.sty`, `slides.sty` and `letter.sty`. These use the command `\@obsoletefile`, which produces a warning message.

```

738 \def\@obsoletefile#1#2{%
739   \@latex@warning@no@line{inputting '#1' instead of obsolete '#2'}}%
740 \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.

```
741 \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.

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

```

@if@listfiles@hashes 743 \ExplSyntaxOn
@if@listfiles@sizes 744 \keys_define:nn { __kernel / listfiles }
745   {
746     hashes .legacy_if_set:n = @listfiles@hashes ,
747     sizes .legacy_if_set:n = @listfiles@sizes
748   }
749 \ExplSyntaxOff

```

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

```

@listfiles 750 \NewDocumentCommand\listfiles{0{} }{%
751   \SetKeys[ __kernel/listfiles ]{#1}%
752   \let\listfiles\relax
753   \def\@listfiles##1##2##3##4##5##6##7##8##9\@{%
754     \def\reserved@d{\ }%
755     \@tfor\reserved@c:=##1##2##3##4##5##6##7##8\do{%
756       \ifx\reserved@c\reserved@d
757         \edef\filename@area{ \filename@area}%
758       \fi}%
759   \def\@dofilelist{%
760     \typeout{^^J *File List*}%
761     \@for\@currname:=\@filelist\do{%
762       \filename@parse@\currname
763       \edef\reserved@a{%
764         \filename@base.%}

```

```

765         \ifx\filename@ext\relax \tex\else\filename@ext\fi}%
766         \expandafter\let\expandafter\reserved@b
767             \csname ver@\reserved@a\endcsname

```

Packages that `\relax` their `\ver@...` string to allow for multiple loading (e.g., `fontenc`) can use `\ver@@...` to store the version information instead.

```

768     \ifx\reserved@b\relax
769         \expandafter\let\expandafter\reserved@b
770             \csname ver@@\reserved@a\endcsname
771         \fi
772         \expandafter\expandafter\expandafter\@listfiles\expandafter
773             \filename@area\filename@base\\\\\\\\\\\\\\\\\\\\\\\\\\@
774         \typeout{%
775             \filename@area\reserved@a
776             \ifx\reserved@b\relax\else\@spaces\reserved@b\fi

```

Now we add the additional information if requested.

```

777     \ifnum0%
778         \if@listfiles@hashes1\fi
779         \if@listfiles@sizes1\fi
780             >0 %
781             ^^J\@spaces
782             (%
783                 \if@listfiles@sizes
784                     size \@dofilelist@size@\currname
785                     \if@listfiles@hashes
786                         ,
787                         \fi
788                     \fi
789                     \if@listfiles@hashes
790                         hash \@dofilelist@hash@\currname
791                     \fi
792                 )%
793                 \fi
794             })%
795             \typeout{ *****^J}}}

```

```

796 \ExplSyntaxOn
797 \cs_new_eq:NN \dofilelist@hash \file_mdfive_hash:n
798 \cs_new_eq:NN \dofilelist@size \file_size:n
799 \ExplSyntaxOff

```

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}
800 \onlypreamble\listfiles

```

```
\@dofilelist 801 \let\@dofilelist\relax  
802 ⟨/2ekernel⟩  
(End of definition for \@obsoletefile and others.)
```

File 21

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 OML, 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 1}
```

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

```
\ProvideTextCommand{\command}{\encoding}{\number}{\default}{\commands}
```

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

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

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

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

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

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

This command declares a text accent. The commands:

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

have the same effect.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\DeclareTextCommandDefault{\command}{\definition}
```

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

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

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

```
\ProvideTextCommandDefault{\command}{\definition}
```

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

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

are short for:

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

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

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

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

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 flaky one and make the default definition point to the new encoding. In such a case defining

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

is not enough since if typesetting in OT1 L^AT_EX will still find the encoding specific definition for OT1 and therefore ignore the new default. Therefore to ensure that in this case the TS1 version is used we have to remove the OT1 declaration:

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

Since the \$ sign is a proper glyph in the T1 encoding there is no point removing its definition and forcing L^AT_EX to pick up the TS1 version if typesetting in this encoding. However, assume you want to use the variant dollar sign, i.e., \$ for your dollars. In that case you have to get rid of the T1 declaration as well, e.g., the following would do that for you:

```
\UndeclareTextCommand{\textdollar}{OT1}
\UndeclareTextCommand{\textdollar} {T1}
\DeclareTextCommandDefault{\textdollar}
  {\UseTextSymbol{TS1}\textdollaroldstyle}
```

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

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.

```
1  {*2ekernel}
2  \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`.

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

```

The command `\@inmathwarn` produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't call `\relax` before the `\ifmmode`. This means that it is possible for the warning to fail to be issued at the beginning of a row of an `\halign` whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a `\relax` would break ligatures and kerning between text symbols.

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

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

(End of definition for `\DeclareTextCommand` and others.)

`\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.

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

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

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

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

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

(End of definition for `\DeclareTextCommandDefault` and `\ProvideTextCommandDefault`.)

`\DeclareTextAccent`

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

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

66 \@onlypreamble\DeclareTextAccent
```

(End of definition for `\DeclareTextAccent`.)

`\add@accent`

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

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

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

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

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

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

```
70   \global\mathchardef\accent@spacefactor\spacefactor}%
```

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 already happened in the box above, if necessary), we simply disable that part of the code temporarily. We also ignore `\ignorespaces` which has the same issue and may show up as part of `\normalfont` if that is used.

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

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

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

(End of definition for `\add@accent`.)

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

(End of definition for `\hmode@bgroup`.)

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

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

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

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

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

```

91 \def\expandafter\noexpand
92     \csname#2\string#\endcsname####1{%
93     \noexpand@text@composite
94         \expandafter\noexpand\csname#2\string#\endcsname
95         ####1\noexpand\empty\noexpand@text@composite
96         {##1}}}}%
97 \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
98 \fi
99 \expandafter\def\csname\expandafter\string\csname
100    #2\endcsname\string#1-\string#3\empty\endcsname{#4}%
101 }
102 
```

(/2ekernel | latexrelease)

- 103 <latexrelease>\EndIncludeInRelease
- 104 <latexrelease>\IncludeInRelease{0000/00/00}{\DeclareTextCompositeCommand}
- 105 <latexrelease> {test for undeclared accent}%
- 106 <latexrelease>\def\DeclareTextCompositeCommand#1#2#3#4{%
- 107 <latexrelease> \expandafter\let\expandafter\reserved@a
- 108 <latexrelease> \csname#2\string#\endcsname
- 109 <latexrelease> \expandafter\expandafter\expandafter\ifx
- 110 <latexrelease> \expandafter\@car\reserved@a\relax\relax\@nil
- 111 <latexrelease> \else
- 112 <latexrelease> \edef\reserved@b##1{%
- 113 <latexrelease> \def\expandafter\noexpand
- 114 <latexrelease> \csname#2\string#\endcsname####1{%
- 115 <latexrelease> \noexpand@text@composite
- 116 <latexrelease> \expandafter\noexpand\csname#2\string#\endcsname
- 117 <latexrelease> ####1\noexpand\empty\noexpand@text@composite
- 118 <latexrelease> {##1}}}}%
- 119 <latexrelease> \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
- 120 <latexrelease> \fi
- 121 <latexrelease> \expandafter\def\csname\expandafter\string\csname
- 122 <latexrelease> #2\endcsname\string#1-\string#3\empty\endcsname{#4}%
- 123 <latexrelease>\EndIncludeInRelease
- 124

(*2ekernel)

125 \onlypreamble\DeclareTextCompositeCommand

This all works because:

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

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

Note that `\@text@\composite` grabs the first token of the argument and puts just that in the csname. This is so that `\'{\textit{e}}` will work—it checks whether `\T1\'-\textit{e}` 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 `\\"{} - \@empty`, which ought to be `\relax`, and so all is well. If we didn't include the `\@empty`, then `\'{}{}` would expand to:

```
\csname \string \T1\` - \string \endcsname
```

so the `\endcsname` would be `\string`ed` and the whole of the rest of the document would be put inside the `\csname`. This would not be good.

```
126 \def\@text@composite#1#2#3\@text@composite{%
127   \expandafter\@text@composite@x
128   \csname\string#1-\string#2\endcsname}
```

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

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

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

```
136 \catcode\z@=11\relax
137 \def\DeclareTextComposite#1#2#3#4{%
138   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
139   \bgroup
140     \lccode\z@#4%
141     \lowercase{%
142       \egroup
143       \reserved@a ^^@}}
144 \catcode\z@=15\relax
145 \onlypreamble\DeclareTextComposite
```

(End of definition for `\DeclareTextCompositeCommand` and others.)

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

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

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

```

150 \DeclareRobustCommand*\UseTextAccent[3]{%
151   \hmode@start@before@group
152   {%
153     \let\hmode@start@before@group\@firstofone
154     \let\@curr@enc\cf@encoding
155     \use@text@encoding{#1}%
156     #2{\use@text@encoding\@curr@enc#3}%
157   }%
158 \DeclareRobustCommand*\UseTextSymbol[2]{%
159   \hmode@start@before@group
160   {%
161     \def\@wrong@font@char{\MessageBreak
162       for \noexpand\symbol`\'{#2}}%
163     \use@text@encoding{#1}%
164     #2%
165   }%
166 }
167 </2ekernel | latexrelease>
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{0000/00/00}%
170 <latexrelease>          {\UseTextAccent}{Make commands robust}%
171 <latexrelease>
172 <latexrelease>\kernel@make@fragile\UseTextAccent
173 <latexrelease>\kernel@make@fragile\UseTextSymbol
174 <latexrelease>
175 <latexrelease>\EndIncludeInRelease
176 <*2ekernel>

```

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

```

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

```

(End of definition for \UseTextAccent, \UseTextSymbol, and \@use@text@encoding.)

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

For a detailed discussion see pr/3160.

```

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

```

(End of definition for \hmode@start@before@group.)

| | |
|---------------------------|--|
| \DeclareTextSymbolDefault | Some syntactic sugar. Again, these should probably be optimized for speed. |
| \DeclareTextAccentDefault | <pre> 185 \def\DeclareTextSymbolDefault#1#2{% 186 \DeclareTextCommandDefault#1{\UseTextSymbol{#2}{#1}}% 187 \def\DeclareTextAccentDefault#1#2{% 188 \DeclareTextCommandDefault#1{\UseTextAccent{#2}{#1}}% 189 \onlypreamble\DeclareTextSymbolDefault 190 \onlypreamble\DeclareTextAccentDefault </pre> <p>(End of definition for <code>\DeclareTextSymbolDefault</code> and <code>\DeclareTextAccentDefault</code>.)</p> |
| \UndeclareTextCommand | <p>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.</p> <pre> 191 \def\UndeclareTextCommand#1#2{% </pre> <p>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).</p> <pre> 192 \expandafter\ifx\csname#2\string#1\endcsname\relax 193 \else </pre> <p>Else: throw away that declaration.</p> <pre> 194 \global\expandafter\let\csname#2\string#1\endcsname 195 \undefined </pre> <p>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 <code>\foo</code> would look similar to <code>\T1-cmd \foo \T1\foo</code> (three tokens).</p> <p>Of course, instead of <code>\T1</code> one could see a different encoding name; which one depends the encoding for which <code>\foo</code> was declared last.</p> <p>Now assume we have just removed the declaration for <code>\foo</code> in <code>\T1</code> and the top-level of <code>\foo</code> expands to the above. Then we better change that pretty fast otherwise we do get an “undefined csname error” when we try to typeset <code>\foo</code> within <code>\T1</code> instead of getting the default definition for <code>\foo</code>. 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 <code>?</code>.</p> <p>Thus in case the last token of the top-level expansion is now undefined we change the declaration to look like <code>\?-cmd \foo \?\foo</code> which is done by the following (readable?) code:</p> <pre> 196 \expandafter\expandafter\expandafter 197 \ifx\expandafter\@thirdofthree#1\@undefined 198 \expandafter\gdef\expandafter#1\expandafter 199 {\csname ?-cmd\expandafter\endcsname\expandafter 200 #1\csname?\string#1\endcsname}% 201 \fi 202 \fi 203 } </pre> <pre> 204 \onlypreamble\UndeclareTextCommand </pre> <p>(End of definition for <code>\UndeclareTextCommand</code>.)</p> |

1.4.2 Hyphenation

```
\patterns
\@@patterns
\hyphenation
\@@hyphenation
205 \%\\let\\@@patterns\\patterns
206 \%\\let\\@@hyphenation\\hyphenation
207 \%\\def\\patterns{%
208 %   \\bgroup
209 %     \\let\\protect\\empty
210 %     \\let\\@typeset\\protect\\empty
211 %     \\let\\@changed@x\\@changed@x@mouth
212 %   \\afterassignment\\egroup
213 %   \\@@patterns
214 %}
215 \%\\def\\hyphenation{%
216 %   \\bgroup
217 %     \\let\\protect\\empty
218 %     \\let\\@typeset\\protect\\empty
219 %     \\let\\@changed@x\\@changed@x@mouth
220 %   \\afterassignment\\egroup
221 %   \\@@hyphenation
222 %}
```

(End of definition for `\patterns` and others.)

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.

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

(End of definition for `\a`.)

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:

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

Some symbols from OT1:

```
240 \% \DeclareTextSymbolDefault{\AA}{OT1}
241 \DeclareTextSymbolDefault{\AE}{OT1}
242 \DeclareTextSymbolDefault{\L}{OT1}
243 \DeclareTextSymbolDefault{\OE}{OT1}
244 \DeclareTextSymbolDefault{\O}{OT1}
245 \% \DeclareTextSymbolDefault{\aa}{OT1}
246 \DeclareTextSymbolDefault{\ae}{OT1}
247 \DeclareTextSymbolDefault{\i}{OT1}
248 \DeclareTextSymbolDefault{\j}{OT1}

249 \DeclareTextSymbolDefault{\ij}{OT1}
250 \DeclareTextSymbolDefault{\IJ}{OT1}

251 \DeclareTextSymbolDefault{\l}{OT1}
252 \DeclareTextSymbolDefault{\oe}{OT1}
253 \DeclareTextSymbolDefault{\o}{OT1}
254 \DeclareTextSymbolDefault{\ss}{OT1}
255 \DeclareTextSymbolDefault{\textdollar}{OT1}
256 \DeclareTextSymbolDefault{\textemdash}{OT1}
257 \DeclareTextSymbolDefault{\textendash}{OT1}
258 \DeclareTextSymbolDefault{\textexclamdown}{OT1}
259 \% \DeclareTextSymbolDefault{\texthphenchar}{OT1}
260 \% \DeclareTextSymbolDefault{\texthphen}{OT1}
261 \DeclareTextSymbolDefault{\textquestiondown}{OT1}
262 \DeclareTextSymbolDefault{\textquotedblleft}{OT1}
263 \DeclareTextSymbolDefault{\textquotedblright}{OT1}
264 \DeclareTextSymbolDefault{\textquotleft}{OT1}
265 \DeclareTextSymbolDefault{\textquotright}{OT1}
266 \DeclareTextSymbolDefault{\textsterling}{OT1}
```

Some symbols from OMS:

```
267 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
268 \DeclareTextSymbolDefault{\textbackslash}{OMS}
269 \DeclareTextSymbolDefault{\textbar}{OMS}
270 \DeclareTextSymbolDefault{\textbardbl}{OMS}
271 \DeclareTextSymbolDefault{\textbraceleft}{OMS}
272 \DeclareTextSymbolDefault{\textbraceright}{OMS}
273 \DeclareTextSymbolDefault{\textbullet}{OMS}
```

```

274 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
275 \DeclareTextSymbolDefault{\textdagger}{OMS}
276 \DeclareTextSymbolDefault{\textparagraph}{OMS}
277 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
278 \DeclareTextSymbolDefault{\textsection}{OMS}
279 \DeclareTextAccentDefault{\textcircled}{OMS}

```

Some symbols from OML:

```

280 \DeclareTextSymbolDefault{\textless}{OML}
281 \DeclareTextSymbolDefault{\textgreater}{OML}
282 \DeclareTextAccentDefault{\t}{OML}

```

Some defaults we can fake.

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

```

283 \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
284 % \expandafter\def\expandafter
285 %           \copyright\expandafter{\expandafter{\copyright}}
286 \DeclareTextCommandDefault{\textasciicircum}{\^{}}
287 \DeclareTextCommandDefault{\textasciitilde}{\~{}}
288 \DeclareTextCommandDefault{\textunderscore}{%
289   \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

```

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

```

290 \% \DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
291 \DeclareTextSymbolDefault{\textcompwordmark}{T1}
292 \DeclareTextCommandDefault{\textvisiblespace}{%
293   \mbox{\kern.06em\vrule\@height.3ex}%
294   \vbox{\hrule\@width.3em}%
295   \hbox{\vrule\@height.3ex}}

```

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

```

296 \DeclareTextCommandDefault{\textellipsis}{%
297   .\kern\fontdimen3\font
298   .\kern\fontdimen3\font
299   .\kern\fontdimen3\font}
300 \% \DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape r}}
301 \DeclareTextCommandDefault{\textregistered}{\textcircled{%
302   \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
303 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
304 \DeclareTextCommandDefault{\SS}{SS}
305 \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
306 \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}

```

1.4.5 Math material

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

```
307 \DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textrm{\$}\fi}
```

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

```
308 \protected\def\{{\ifmmode\lbrace\else\textrm{\{}fi\}}
309 \protected\def\}{\ifmmode\rbrace\else\textrm{\}}fi\}
310 \DeclareRobustCommand{\P}{\ifmmode\mathparagraph\else\textrm{\P}\fi}
311 \DeclareRobustCommand{\S}{\ifmmode\mathsection\else\textrm{\S}\fi}
312 \DeclareRobustCommand{\dag}{\ifmmode\dagger\else\textrm{\dag}\fi}
313 \DeclareRobustCommand{\ddag}{\ifmmode\ddagger\else\textrm{\ddagger}\fi}
```

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

```
314 \DeclareRobustCommand{\_}{%
315   \ifmmode\nfss@text{\textunderscore}\else\textrm{\_}\fi}
316 \DeclareRobustCommand{\copyright}{%
317   \ifmmode\nfss@text{\textcopyright}\else\textrm{\copyright}\fi}
318 \DeclareRobustCommand{\pounds}{%
319   \ifmmode\mathsterling\else\textrm{\pounds}\fi}
320 \DeclareRobustCommand{\dots}{%
321   \ifmmode\mathellipsis\else\textrm{\dots}\fi}
322 \let\ldots\dots
```

Default definition of the commabelow accent.

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

```

350  \let\csname\string\T1\string\c-r\endcsname\@undefined
351  \let\csname\string\T1\string\c-r\endcsname\@undefined
352  \EndIncludeInRelease
      Default definition of the commaabove accent(E.G.).
353  \IncludeInRelease{2016/02/01}{\textcommablock}{comma above}%
354  {*2ekernel | latexrelease}
355  \DeclareTextCommandDefault{\textcommablock}[1]{%
356    \hmode@bgroup
357    \oalign{%
358      \hidewidth
359      \raise.7ex\hbox{%
360        \check@mathfonts\fontsize\ssf@size\z@\math@fontsfalse\selectfont'%
361      }%
362      \hidewidth\crcr
363      \null#1\crcr
364    }%
365    \egroup
366  }
367  \EndIncludeInRelease
368  {/2ekernel | latexrelease}
369  \IncludeInRelease{0000/00/00}{\textcommablock}{comma above}%
370  \let\textcommablock\@undefined
371  \expandafter
372  \let\csname\string\OT1\string\c-g\endcsname\@undefined
373  \expandafter
374  \let\csname\string\T1\string\c-g\endcsname\@undefined
375  \EndIncludeInRelease

```

1.5 Definitions for the OT1 encoding

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

Declare the encoding.

```

376  {*OT1}
377  \DeclareFontEncoding{OT1}{}{}

```

Declare the accents.

```

378  \DeclareTextAccent{"}{OT1}{127}
379  \DeclareTextAccent{'}{OT1}{19}
380  \DeclareTextAccent{.}{OT1}{95}
381  \DeclareTextAccent{=}{OT1}{22}
382  \DeclareTextAccent{`}{OT1}{94}
383  \DeclareTextAccent{'}{OT1}{18}
384  \DeclareTextAccent{`~}{OT1}{126}
385  \DeclareTextAccent{H}{OT1}{125}
386  \DeclareTextAccent{u}{OT1}{21}
387  \DeclareTextAccent{v}{OT1}{20}
388  \DeclareTextAccent{r}{OT1}{23}

```

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

```

389  \DeclareTextCommand{\b}{OT1}[1]
390    {\hmode@bgroup\o@align{\relax#1\crcr\hidewidth\ltx@sh@ft{-3ex}}%

```

```

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

```

Declare the text symbols.

```

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

```

The `\nobreak\hspace{z@}` is there to prevent a break after the hyphen but allow later breaks in the remainder of the word.

```

409 \DeclareTextCommand{\textnonbreakinghyphen}{OT1}{\mbox{-}\nobreak\hspace{z@}}
410 \DeclareTextCommand{\textfiguredash} {OT1}{\textendash}
411 \DeclareTextCommand{\texthorizontalbar} {OT1}{\textemdash}

```

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

```

412 %\DeclareTextSymbol{\textexclamdown}{OT1}{60}
413 %\DeclareTextSymbol{\textquestiondown}{OT1}{62}
414 \DeclareTextCommand{\textexclamdown}{OT1}{!`}
415 \DeclareTextCommand{\textquestiondown}{OT1}{?`}
416 %\DeclareTextSymbol{\texthyphenchar}{OT1}{`-}
417 %\DeclareTextSymbol{\texthyphen}{OT1}{`-}
418 \DeclareTextSymbol{\textquotedblleft}{OT1}{92}
419 \DeclareTextSymbol{\textquotedblright}{OT1}{`}
420 \DeclareTextSymbol{\textquotelleft}{OT1}{`}
421 \DeclareTextSymbol{\textquoteright}{OT1}{`}

```

Some symbols which are faked from others:

```

422 % \DeclareTextCommand{\aa}{OT1}
423 %   {{\accent23a}}
424 \DeclareTextCommand{\L}{OT1}
425   {\leavevmode\setbox\z@\hbox{L}\hb@xt@wd\z@{\hss@xxxii L}}
426 \DeclareTextCommand{\l}{OT1}
427   {\hmode@bgroup\@xxxii l\egroup}
428 % \DeclareTextCommand{\AA}{OT1}
429 %   {\leavevmode\setbox\z@\hbox{h}\dimen@ht\z@\advance\dimen@-1ex%
430 %   \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`.

```

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

```

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

```

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

```

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

```

438 \DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
439   \ifdim \fontdimen\@ne\font >\z@
440     \slshape
441   \else
442     \upshape
443   \fi
444   \char`\$\egroup}
445 \DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
446   \ifdim \fontdimen\@ne\font >\z@
447     \itshape
448   \else
449     \fontshape{ui}\selectfont
450   \fi
451   \char`\$\egroup}

```

Here we are adding some more composite commands to the OT1 encoding. This makes the use of certain accents with i compatible with their use with the T1 encoding; this enables them to become true L^AT_EX internal representations. However, it will make these accents work a little less fast since a check will always be made for the existence of a composite.

```

452 \DeclareTextComposite{\.}{OT1}{i}{`i}
453 \DeclareTextComposite{\.}{OT1}{i}{`i}
454 \DeclareTextCompositeCommand{\`}{OT1}{i}{\@tabacckludge`i}
455 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'i}
456 \DeclareTextCompositeCommand{\^}{OT1}{i}{^\i}
457 \DeclareTextCompositeCommand{\"}{OT1}{i}{\"i}

```

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

```

458 \ifx\textcommaabove\undefined\else
459 \DeclareTextCompositeCommand{\c}{OT1}{g}{\textcommaabove{g}}
460 \fi
461 
```

1.6 Definitions for the T1 encoding

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

Declare the encoding.

```

462 <*T1>
463 \DeclareFontEncoding{T1}{}{}

```

Declare the accents.

```

464 \DeclareTextAccent{\`}{T1}{0}
465 \DeclareTextAccent{\'}{T1}{1}
466 \DeclareTextAccent{\^}{T1}{2}

```

```

467 \DeclareTextAccent{\~}{T1}{3}
468 \DeclareTextAccent{\^}{T1}{4}
469 \DeclareTextAccent{\H}{T1}{5}
470 \DeclareTextAccent{\r}{T1}{6}
471 \DeclareTextAccent{\v}{T1}{7}
472 \DeclareTextAccent{\u}{T1}{8}
473 \DeclareTextAccent{\=}{T1}{9}
474 \DeclareTextAccent{\.}{T1}{10}

```

Some accents have to be built by hand. Note that `\o@align` 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.

```

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

```

Some symbols are constructed.

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

```

490 \DeclareTextCommand{\textperthousand}{T1}
491   {\%\char 24 } % space or 'relax as delimiter?
492 \DeclareTextCommand{\textpertenthousand}{T1}
493   {\%\char 24\char 24 } % space or 'relax as delimiter?

```

For Maltese, `\Hwithstroke` and `\hwithstroke` are needed.

```

494 \DeclareTextCommand{\Hwithstroke}{T1}
495   {%
496     \hmode@bgroup
497     \vphantom{H}%
498     \sbox\z@{H}%
499     \o@align{%
500       H\cr
501       \hidewidth
502       \vrule
503         height \dimexpr 0.7\ht\z@+0.1ex\relax
504         depth -0.7\ht\z@
505         width 0.8\wd\z@
506       \hidewidth\cr
507     }%
508     \egroup
509   }
510 \DeclareTextCommand{\hwithstroke}{T1}
511   {%

```

```

512     \hmode@bgroup
513     \vphantom{h}%
514     \sbox{z@{h}}%
515     \ooalign{%
516         h\cr
517         \kern0.075\wd{z@}
518         \vrule
519             height \dimexpr 0.7\ht{z@}+0.1ex\relax
520             depth -0.7\ht{z@}
521             width 0.4\wd{z@}
522             \hidewidth\cr
523     }%
524     \egroup
525 }
```

Declare the text symbols.

```

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

540 \ DeclareTextSymbol{\guillemetleft}{T1}{19}
541 \ DeclareTextSymbol{\guillemetright}{T1}{20}
542 % old Adobe names
543 \ DeclareTextSymbol{\guillemotleft}{T1}{19}
544 \ DeclareTextSymbol{\guillemotright}{T1}{20}

545 \ DeclareTextSymbol{\guilsinglleft}{T1}{14}
546 \ DeclareTextSymbol{\guilsinglright}{T1}{15}
547 \ DeclareTextSymbol{\i}{T1}{25}
548 \ DeclareTextSymbol{\j}{T1}{26}
549 \ DeclareTextSymbol{\ij}{T1}{188}
550 \ DeclareTextSymbol{\IJ}{T1}{156}
551 \ DeclareTextSymbol{\l}{T1}{170}
552 \ DeclareTextSymbol{\ng}{T1}{173}
553 \ DeclareTextSymbol{\oe}{T1}{247}
554 \ DeclareTextSymbol{\o}{T1}{248}
555 \ DeclareTextSymbol{\quotedblbase}{T1}{18}
556 \ DeclareTextSymbol{\quotesinglbase}{T1}{13}
557 \ DeclareTextSymbol{\ss}{T1}{255}
558 \ DeclareTextSymbol{\textasciicircum}{T1}{`^}
559 \ DeclareTextSymbol{\textasciitilde}{T1}{`~}
560 \ DeclareTextSymbol{\textbackslash}{T1}{``\\`}
561 \ DeclareTextSymbol{\textbar}{T1}{`|`}
562 \ DeclareTextSymbol{\textbraceleft}{T1}{`{|`}
```

```

563 \DeclareTextSymbol{\textbraceright}{T1}{`}
564 \DeclareTextSymbol{\textcompwordmark}{T1}{23}
565 \DeclareTextSymbol{\textdollar}{T1}{`}
566 \DeclareTextSymbol{\textemdash}{T1}{22}
567 \DeclareTextSymbol{\textendash}{T1}{21}

```

The `\nobreak\hskip\z@` is there to prevent a break after the hyphen but allow later breaks in the remainder of the word.

```

568 \DeclareTextCommand{\textnonbreakinghyphen}{T1}{\mbox{-}\nobreak\hskip\z@}
569 \DeclareTextCommand{\textfiguredash} {T1}{\textendash}
570 \DeclareTextCommand{\texthorizontalbar} {T1}{\textemdash}

571 \DeclareTextSymbol{\textexclamdown}{T1}{189}
572 \DeclareTextSymbol{\textgreater}{T1}{`}
573 %\DeclareTextSymbol{\texthyphenchar}{T1}{127}
574 %\DeclareTextSymbol{\texthyphen}{T1}{`}
575 \DeclareTextSymbol{\textless}{T1}{`}
576 \DeclareTextSymbol{\textquestiondown}{T1}{190}
577 \DeclareTextSymbol{\textquotedblleft}{T1}{16}
578 \DeclareTextSymbol{\textquotedblright}{T1}{17}
579 \DeclareTextSymbol{\textquotedbl}{T1}{`}
580 \DeclareTextSymbol{\textquotelleft}{T1}{`}
581 \DeclareTextSymbol{\textquoteright}{T1}{`}
582 \DeclareTextSymbol{\textsection}{T1}{159}
583 \DeclareTextSymbol{\textsterling}{T1}{191}
584 \DeclareTextSymbol{\textunderscore}{T1}{95}
585 \DeclareTextSymbol{\textvisiblespace}{T1}{32}
586 \DeclareTextSymbol{\th}{T1}{254}

```

Declare the composites.

```

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

"80 = 128
589 \DeclareTextComposite{\.{u}}{T1}{A}{128}
590 \DeclareTextComposite{\.{k}}{T1}{A}{129}
591 \DeclareTextComposite{\.{v}}{T1}{C}{130}
592 \DeclareTextComposite{\.{v}}{T1}{C}{131}
593 \DeclareTextComposite{\.{v}}{T1}{D}{132}
594 \DeclareTextComposite{\.{v}}{T1}{E}{133}
595 \DeclareTextComposite{\.{k}}{T1}{E}{134}
596 \DeclareTextComposite{\.{u}}{T1}{G}{135}

"88 = 136
597 \DeclareTextComposite{\.{v}}{T1}{L}{136}
598 \DeclareTextComposite{\.{v}}{T1}{L}{137}
599 \DeclareTextComposite{\.{v}}{T1}{N}{139}
600 \DeclareTextComposite{\.{v}}{T1}{N}{140}
601 \DeclareTextComposite{\.{H}}{T1}{O}{142}
602 \DeclareTextComposite{\.{v}}{T1}{R}{143}

"90 = 144
603 \DeclareTextComposite{\.{v}}{T1}{R}{144}
604 \DeclareTextComposite{\.{v}}{T1}{S}{145}
605 \DeclareTextComposite{\.{v}}{T1}{S}{146}
606 \DeclareTextComposite{\.{c}}{T1}{S}{147}

```

```

607 \DeclareTextComposite{\v}{T1}{T}{148}
608 \DeclareTextComposite{\c}{T1}{T}{149}
609 \DeclareTextComposite{\H}{T1}{U}{150}
610 \DeclareTextComposite{\r}{T1}{U}{151}

"A8 = 152
611 \DeclareTextComposite{\"}{T1}{Y}{152}
612 \DeclareTextComposite{\'}{T1}{Z}{153}
613 \DeclareTextComposite{\v}{T1}{Z}{154}
614 \DeclareTextComposite{\.}{T1}{Z}{155}
615 \DeclareTextComposite{\.}{T1}{I}{157}

"A0 = 160
616 \DeclareTextComposite{\u}{T1}{a}{160}
617 \DeclareTextComposite{\k}{T1}{a}{161}
618 \DeclareTextComposite{\'}{T1}{c}{162}
619 \DeclareTextComposite{\v}{T1}{c}{163}
620 \DeclareTextComposite{\v}{T1}{d}{164}
621 \DeclareTextComposite{\v}{T1}{e}{165}
622 \DeclareTextComposite{\k}{T1}{e}{166}
623 \DeclareTextComposite{\u}{T1}{g}{167}

"A8 = 168
624 \DeclareTextComposite{\'}{T1}{l}{168}
625 \DeclareTextComposite{\v}{T1}{l}{169}
626 \DeclareTextComposite{\'}{T1}{n}{171}
627 \DeclareTextComposite{\v}{T1}{n}{172}
628 \DeclareTextComposite{\H}{T1}{o}{174}
629 \DeclareTextComposite{\'}{T1}{r}{175}

"B0 = 176
630 \DeclareTextComposite{\v}{T1}{r}{176}
631 \DeclareTextComposite{\'}{T1}{s}{177}
632 \DeclareTextComposite{\v}{T1}{s}{178}
633 \DeclareTextComposite{\c}{T1}{s}{179}
634 \DeclareTextComposite{\v}{T1}{t}{180}
635 \DeclareTextComposite{\c}{T1}{t}{181}
636 \DeclareTextComposite{\H}{T1}{u}{182}
637 \DeclareTextComposite{\r}{T1}{u}{183}

"B8 = 184
638 \DeclareTextComposite{\"}{T1}{y}{184}
639 \DeclareTextComposite{\'}{T1}{z}{185}
640 \DeclareTextComposite{\v}{T1}{z}{186}
641 \DeclareTextComposite{\.}{T1}{z}{187}

"C0 = 192
642 \DeclareTextComposite{\'}{T1}{A}{192}
643 \DeclareTextComposite{\'}{T1}{A}{193}
644 \DeclareTextComposite{\^}{T1}{A}{194}
645 \DeclareTextComposite{\~}{T1}{A}{195}
646 \DeclareTextComposite{\"}{T1}{A}{196}
647 \DeclareTextComposite{\r}{T1}{A}{197}
648 \DeclareTextComposite{\c}{T1}{C}{199}

```

```

"C8 = 200
649 \DeclareTextComposite{\`}{T1}{E}{200}
650 \DeclareTextComposite{\'}{T1}{E}{201}
651 \DeclareTextComposite{\^}{T1}{E}{202}
652 \DeclareTextComposite{\"}{T1}{E}{203}
653 \DeclareTextComposite{\`}{T1}{I}{204}
654 \DeclareTextComposite{\'}{T1}{I}{205}
655 \DeclareTextComposite{\^}{T1}{I}{206}
656 \DeclareTextComposite{\\"}{T1}{I}{207}

"D0 = 208
657 \DeclareTextComposite{\~}{T1}{N}{209}
658 \DeclareTextComposite{\'}{T1}{O}{210}
659 \DeclareTextComposite{\'}{T1}{O}{211}
660 \DeclareTextComposite{\^}{T1}{O}{212}
661 \DeclareTextComposite{\~}{T1}{O}{213}
662 \DeclareTextComposite{\\"}{T1}{O}{214}

"D8 = 216
663 \DeclareTextComposite{\`}{T1}{U}{217}
664 \DeclareTextComposite{\'}{T1}{U}{218}
665 \DeclareTextComposite{\^}{T1}{U}{219}
666 \DeclareTextComposite{\\"}{T1}{U}{220}
667 \DeclareTextComposite{\'}{T1}{Y}{221}

"E0 = 224
668 \DeclareTextComposite{\`}{T1}{a}{224}
669 \DeclareTextComposite{\'}{T1}{a}{225}
670 \DeclareTextComposite{\^}{T1}{a}{226}
671 \DeclareTextComposite{\~}{T1}{a}{227}
672 \DeclareTextComposite{\\"}{T1}{a}{228}
673 \DeclareTextComposite{\r}{T1}{a}{229}
674 \DeclareTextComposite{\c}{T1}{c}{231}

"E8 = 232
675 \DeclareTextComposite{\`}{T1}{e}{232}
676 \DeclareTextComposite{\'}{T1}{e}{233}
677 \DeclareTextComposite{\^}{T1}{e}{234}
678 \DeclareTextComposite{\\"}{T1}{e}{235}
679 \DeclareTextComposite{\'}{T1}{i}{236}
680 \DeclareTextComposite{\`}{T1}{i}{236}
681 \DeclareTextComposite{\'}{T1}{i}{237}
682 \DeclareTextComposite{\'}{T1}{i}{237}
683 \DeclareTextComposite{\^}{T1}{i}{238}
684 \DeclareTextComposite{\^}{T1}{i}{238}
685 \DeclareTextComposite{\\"}{T1}{i}{239}
686 \DeclareTextComposite{\\"}{T1}{i}{239}

"F0 = 240
687 \DeclareTextComposite{\~}{T1}{n}{241}
688 \DeclareTextComposite{\'}{T1}{o}{242}
689 \DeclareTextComposite{\'}{T1}{o}{243}
690 \DeclareTextComposite{\^}{T1}{o}{244}
691 \DeclareTextComposite{\~}{T1}{o}{245}
692 \DeclareTextComposite{\\"}{T1}{o}{246}

```

```

" F8 = 248

693 \DeclareTextComposite{\`}{T1}{u}{249}
694 \DeclareTextComposite{\'}{T1}{u}{250}
695 \DeclareTextComposite{\^}{T1}{u}{251}
696 \DeclareTextComposite{\"}{T1}{u}{252}
697 \DeclareTextComposite{\'}{T1}{y}{253}

698 \DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
699 \DeclareTextCompositeCommand{\k}{T1}{O}{\textogonekcentered{O}}

700 \ifx\textcommaabove\@undefined\else
701 \DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
702 \fi
703 \ifx\textcommabelow\@undefined\else
704 \DeclareTextCompositeCommand{\c}{T1}{G}{\textcommabelow{G}}
705 \DeclareTextCompositeCommand{\c}{T1}{K}{\textcommabelow{K}}
706 \DeclareTextCompositeCommand{\c}{T1}{k}{\textcommabelow{k}}
707 \DeclareTextCompositeCommand{\c}{T1}{L}{\textcommabelow{L}}
708 \DeclareTextCompositeCommand{\c}{T1}{l}{\textcommabelow{l}}
709 \DeclareTextCompositeCommand{\c}{T1}{N}{\textcommabelow{N}}
710 \DeclareTextCompositeCommand{\c}{T1}{n}{\textcommabelow{n}}
711 \DeclareTextCompositeCommand{\c}{T1}{R}{\textcommabelow{R}}
712 \DeclareTextCompositeCommand{\c}{T1}{r}{\textcommabelow{r}}
713 \fi
714 

```

1.7 Definitions for the OMS encoding

The definitions for the ‘ T_{EX} math symbol’ (OMS) encoding. Even though this is meant to be a math font, it includes some of the standard L^AT_EX text symbols.

Declare the encoding.

```

715 {*OMS}
716 \DeclareFontEncoding{OMS}{}{}

```

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

```

717 \DeclareTextSymbol{\textasteriskcentered}{OMS}{3} % "03
718 \DeclareTextSymbol{\textbackslash}{OMS}{110} % "6E
719 \DeclareTextSymbol{\textbar}{OMS}{106} % "6A
720 \DeclareTextSymbol{\textbardbl}{OMS}{107} % "6B
721 \DeclareTextSymbol{\textbraceleft}{OMS}{102} % "66
722 \DeclareTextSymbol{\textbraceright}{OMS}{103} % "67
723 \DeclareTextSymbol{\textbullet}{OMS}{15} % "0F
724 \DeclareTextSymbol{\textdaggerdbl}{OMS}{122} % "7A
725 \DeclareTextSymbol{\textdagger}{OMS}{121} % "79
726 \DeclareTextSymbol{\textparagraph}{OMS}{123} % "7B
727 \DeclareTextSymbol{\textperiodcentered}{OMS}{1} % "01
728 \DeclareTextSymbol{\textsection}{OMS}{120} % "78
729 \DeclareTextSymbol{\textbigcircle}{OMS}{13} % "0D
730 \DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
731   \ooalign{\%
732     \hfil\raise .07ex\hbox {\upshape#1}\hfil \crcr
733     \char 13 % "0D

```

```

734   }%
735   \egroup}
736 
```

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.

```

737 <*OML>
738 \DeclareFontEncoding{OML}{}{}

```

Declare the symbols.

```

739 \DeclareTextSymbol{\textless}{OML}{`<}
740 \DeclareTextSymbol{\textgreater}{OML}{`>}
741 \DeclareTextAccent{\t}{OML}{127} % "7F
742 
```

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.

```

743 <*OT4>
744 \DeclareFontEncoding{OT4}{}{}
745 \DeclareFontSubstitution{OT4}{cmr}{m}{n}

```

Declare the accents.

```

746 \DeclareTextAccent{\"}{OT4}{127}
747 \DeclareTextAccent{\'}{OT4}{19}
748 \DeclareTextAccent{\.}{OT4}{95}
749 \DeclareTextAccent{\=}{OT4}{22}
750 \DeclareTextAccent{\^}{OT4}{94}
751 \DeclareTextAccent{\' }{OT4}{18}
752 \DeclareTextAccent{\~}{OT4}{126}
753 \DeclareTextAccent{\H}{OT4}{125}
754 \DeclareTextAccent{\u}{OT4}{21}
755 \DeclareTextAccent{\v}{OT4}{20}
756 \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:

```

757 \DeclareTextCommand{\k}{OT4}[1]{%
758   \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.

```

759 \DeclareTextCommand{\b}{OT4}[1]
760   {\hmode@bgroup\o@align{\relax#1\crcr\hidewidth\ltx@sh@ft{-3ex}%
761     \vbox to .2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
762 \DeclareTextCommand{\c}{OT4}[1]
763   {\leavevmode\setbox\z@\hbox{\#1}\ifdim\ht\z@=1ex\accent24 #1%
764     \else\ooalign{\unhbox\z@\crcr\hidewidth\char24\hidewidth}\fi}
765 \DeclareTextCommand{\d}{OT4}[1]
766   {\hmode@bgroup
767     \o@align{\relax#1\crcr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

768 \DeclareTextSymbol{\AE}{OT4}{29}
769 \DeclareTextSymbol{\OE}{OT4}{30}
770 \DeclareTextSymbol{\O}{OT4}{31}
771 \DeclareTextSymbol{\L}{OT4}{138}
772 \DeclareTextSymbol{\ae}{OT4}{26}

773 \DeclareTextSymbol{\guillemetleft}{OT4}{174}
774 \DeclareTextSymbol{\guillemetright}{OT4}{175}
775 % old Adobe names
776 \DeclareTextSymbol{\guillemotleft}{OT4}{174}
777 \DeclareTextSymbol{\guillemotright}{OT4}{175}

778 \DeclareTextSymbol{\i}{OT4}{16}
779 \DeclareTextSymbol{\j}{OT4}{17}
780 \DeclareTextSymbol{\l}{OT4}{170}
781 \DeclareTextSymbol{\o}{OT4}{28}
782 \DeclareTextSymbol{\oe}{OT4}{27}
783 \DeclareTextSymbol{\quotedblbase}{OT4}{255}
784 \DeclareTextSymbol{\ss}{OT4}{25}
785 \DeclareTextSymbol{\textemdash}{OT4}{124}
786 \DeclareTextSymbol{\textendash}{OT4}{123}
787 \DeclareTextSymbol{\textexclamdown}{OT4}{60}
788 \% \DeclareTextSymbol{\texthyphenchar}{OT4}{`-}
789 \% \DeclareTextSymbol{\texthyphen}{OT4}{`-}
790 \DeclareTextSymbol{\textquestiondown}{OT4}{62}
791 \DeclareTextSymbol{\textquotedblleft}{OT4}{92}
792 \DeclareTextSymbol{\textquotedblright}{OT4}{`}
793 \DeclareTextSymbol{\textquotel}{OT4}{`}
794 \DeclareTextSymbol{\textquoter}{OT4}{`}

```

Definition for Å as in OT1:

```

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

```

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

```

798 \DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
799   \ifdim \fontdimen\onefont >\z@
800     \slshape
801   \else
802     \upshape
803   \fi
804   \char`\$\egroup}

```

```

805 \DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
806   \ifdim \fontdimen\@ne\font >\z@
807     \itshape
808   \else
809     \fontshape{ui}\selectfont
810   \fi
811 \char`\$\egroup}

```

Declare the composites.

```

812 \DeclareTextComposite{\k}{OT4}{A}{129}
813 \DeclareTextComposite{\'}{OT4}{C}{130}
814 \DeclareTextComposite{\k}{OT4}{E}{134}
815 \DeclareTextComposite{\'}{OT4}{N}{139}
816 \DeclareTextComposite{\'}{OT4}{S}{145}
817 \DeclareTextComposite{\'}{OT4}{Z}{153}
818 \DeclareTextComposite{\.}{OT4}{Z}{155}
819 \DeclareTextComposite{\k}{OT4}{a}{161}
820 \DeclareTextComposite{\'}{OT4}{c}{162}
821 \DeclareTextComposite{\k}{OT4}{e}{166}
822 \DeclareTextComposite{\'}{OT4}{n}{171}
823 \DeclareTextComposite{\'}{OT4}{s}{177}
824 \DeclareTextComposite{\'}{OT4}{z}{185}
825 \DeclareTextComposite{\.}{OT4}{z}{187}
826 \DeclareTextComposite{\'}{OT4}{o}{211}
827 \DeclareTextComposite{\'}{OT4}{o}{243}
828 
```

1.10 Definitions for the TS1 encoding

```

829 <*TS1>
830 \DeclareFontEncoding{TS1}{}{}
831 \DeclareFontSubstitution{TS1}{cmr}{m}{n}

```

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

```

832 \DeclareTextCommand{\capitalcedilla}{TS1}[1]
833   {\hmode@bgroup
834   \ooalign{\null#1\crcr\hidewidth\char11\hidewidth}\egroup}
835 \DeclareTextCommand{\capitalogonek}{TS1}[1]
836   {\hmode@bgroup
837   \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

```

838 \DeclareTextAccent{\capitalgrave}{TS1}{0}
839 \DeclareTextAccent{\capitalacute}{TS1}{1}
840 \DeclareTextAccent{\capitalcircumflex}{TS1}{2}
841 \DeclareTextAccent{\capitaltilde}{TS1}{3}
842 \DeclareTextAccent{\capitaldieresis}{TS1}{4}
843 \DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}

```

```

844 \DeclareTextAccent{\capitalring}{TS1}{6}
845 \DeclareTextAccent{\capitalcaron}{TS1}{7}
"08 = 8
846 \DeclareTextAccent{\capitalbreve}{TS1}{8}
847 \DeclareTextAccent{\capitalmacron}{TS1}{9}
848 \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 asymmetric 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.

```

" = 
849 \DeclareTextAccent{\t}{TS1}{26}
850 \DeclareTextAccent{\capitaltie}{TS1}{27}
851 \DeclareTextAccent{\newtie}{TS1}{28}
852 \DeclareTextAccent{\capitalnewtie}{TS1}{29}

```

Compound word marks.

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

```

853 \DeclareTextSymbol{\textcapitalcompwordmark}{TS1}{23}
854 \DeclareTextSymbol{\textascendercompwordmark}{TS1}{31}

```

The text companion symbols.

```

855 \DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
"10 = 16
856 \DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
857 \DeclareTextSymbol{\texttwelvedash}{TS1}{21}
858 \DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
"18 = 24
859 \DeclareTextSymbol{\textleftarrow}{TS1}{24}
860 \DeclareTextSymbol{\textrightarrow}{TS1}{25}
"20 = 32
861 \DeclareTextSymbol{\textblank}{TS1}{32}
862 \DeclareTextSymbol{\textdollar}{TS1}{36}
863 \DeclareTextSymbol{\textquotesingle}{TS1}{39}
"28 = 40

```

The symbol `\textasteriskcentered` “*” is supposed to be always available in `TS1` and that is important as it is used in footnote symbols. However, in a few fonts it is missing even though they are otherwise fairly complete. We therefore use a rather elaborate method and check if the slot has a glyph and if not produce a poor man’s version by using a normal “*” slightly enlarged and somewhat lowered. The main application for this symbol is in footnote symbols and there it should produce a comparable size and show a similar placement.

```

864 \% \DeclareTextSymbol{\textasteriskcentered}{TS1}{42} % that's wanted
865 \DeclareTextCommand \textasteriskcentered{TS1}{% % and that's needed
866   \iffontchar\font 42 \char42 \else
867     \begingroup\fontencoding{T1}%
868       \fontsize

```

```

869      {\the\dimexpr1.3\dimexpr\f@size pt\relax}%
870      {\f@baselineskip}%
871      \selectfont
872      \raisebox{-0.7ex}{[\dimexpr\height-0.7ex][0pt]{*}}%
873      \endgroup
874      \fi
875 }

Note that '054 is a comma and '056 is a full stop: these make numbers using oldstyle digits easier to input.

876 \DeclareTextSymbol{\textdblhyphen}{TS1}{45}
877 \DeclareTextSymbol{\textfractionsolidus}{TS1}{47}

Oldstyle digits.

'30 = 48

878 \DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
879 \DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
880 \DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
881 \DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
882 \DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
883 \DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
884 \DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
885 \DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}

'38 = 56

886 \DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
887 \DeclareTextSymbol{\textnineoldstyle}{TS1}{57}

More text companion symbols.

888 \DeclareTextSymbol{\textlangel}{TS1}{60}
889 \DeclareTextSymbol{\textminus}{TS1}{61}
890 \DeclareTextSymbol{\textrangle}{TS1}{62}

'48 = 72

891 \DeclareTextSymbol{\textmho}{TS1}{77}

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

892 \DeclareTextSymbol{\textbigcircle}{TS1}{79}
893 \DeclareTextCommand{\textcircled}{TS1}[1]{\hmode@bgroup
894   \oalign{%
895     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
896     \char 79 % '117 = "4F
897   }%
898 \egroup}

More text companion symbols.

'50 = 80

899 \DeclareTextSymbol{\textohm}{TS1}{87}

'58 = 88

900 \DeclareTextSymbol{\textlbrackdbl}{TS1}{91}
901 \DeclareTextSymbol{\textrbrackdbl}{TS1}{93}
902 \DeclareTextSymbol{\textuparrow}{TS1}{94}
903 \DeclareTextSymbol{\textdownarrow}{TS1}{95}

```

```

"60 = 96
904 \DeclareTextSymbol{\textasciigrave}{TS1}{96}
905 \DeclareTextSymbol{\textborn}{TS1}{98}
906 \DeclareTextSymbol{\textdivorced}{TS1}{99}
907 \DeclareTextSymbol{\textdied}{TS1}{100}

"68 = 104
908 \DeclareTextSymbol{\textleaf}{TS1}{108}
909 \DeclareTextSymbol{\textmarried}{TS1}{109}
910 \DeclareTextSymbol{\textmusicalnote}{TS1}{110}

"78 = 120
911 \DeclareTextSymbol{\texttildelow}{TS1}{126}
This glyph, \textdblhyphenchar is hanging, like the hyphenchar of the ec fonts.
912 \DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}

"80 = 128
913 \DeclareTextSymbol{\textasciibreve}{TS1}{128}
914 \DeclareTextSymbol{\textasciicaron}{TS1}{129}
This next glyph is not the same as \textquotedbl.
915 \DeclareTextSymbol{\textacutedbl}{TS1}{130}
916 \DeclareTextSymbol{\textgravedbl}{TS1}{131}
917 \DeclareTextSymbol{\textdagger}{TS1}{132}
918 \DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
919 \DeclareTextSymbol{\textbardbl}{TS1}{134}
920 \DeclareTextSymbol{\textperthousand}{TS1}{135}

"88 = 136
921 \DeclareTextSymbol{\textbullet}{TS1}{136}
922 \DeclareTextSymbol{\textcelsius}{TS1}{137}
923 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
924 \DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
925 \DeclareTextSymbol{\textflorin}{TS1}{140}
926 \DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
927 \DeclareTextSymbol{\textwon}{TS1}{142}
928 \DeclareTextSymbol{\textnaira}{TS1}{143}

"90 = 144
929 \DeclareTextSymbol{\textguarani}{TS1}{144}
930 \DeclareTextSymbol{\textpeso}{TS1}{145}
931 \DeclareTextSymbol{\textlira}{TS1}{146}
932 \DeclareTextSymbol{\textrecipe}{TS1}{147}
933 \DeclareTextSymbol{\textinterrobang}{TS1}{148}
934 \DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
935 \DeclareTextSymbol{\textdong}{TS1}{150}
936 \DeclareTextSymbol{\texttrademark}{TS1}{151}

"98 = 152
937 \DeclareTextSymbol{\textpertenthousand}{TS1}{152}
938 \DeclareTextSymbol{\textpilcrow}{TS1}{153}
939 \DeclareTextSymbol{\textbaht}{TS1}{154}
940 \DeclareTextSymbol{\textnumero}{TS1}{155}

```

This next name may change. For the following sign we know only a german name, which is abziglich. The meaning is something like “commercial minus”. An ASCII ersatz is ./ (dot slash dot). The temporary English name is \textdiscount.

```

941 \DeclareTextSymbol{\textdiscount}{TS1}{156}
942 \DeclareTextSymbol{\textestimated}{TS1}{157}
943 \DeclareTextSymbol{\textopenbullet}{TS1}{158}
944 \DeclareTextSymbol{\textservicemark}{TS1}{159}

"A0 = 160
945 \DeclareTextSymbol{\textlquill}{TS1}{160}
946 \DeclareTextSymbol{\textrquill}{TS1}{161}
947 \DeclareTextSymbol{\textcent}{TS1}{162}
948 \DeclareTextSymbol{\textsterling}{TS1}{163}
949 \DeclareTextSymbol{\textcurrency}{TS1}{164}
950 \DeclareTextSymbol{\textyen}{TS1}{165}
951 \DeclareTextSymbol{\textbrokenbar}{TS1}{166}
952 \DeclareTextSymbol{\textsection}{TS1}{167}

"A8 = 168
953 \DeclareTextSymbol{\textasciidieresis}{TS1}{168}
954 \DeclareTextSymbol{\textcopyright}{TS1}{169}
955 \DeclareTextSymbol{\textordfeminine}{TS1}{170}
956 \DeclareTextSymbol{\textcopyleft}{TS1}{171}
957 \DeclareTextSymbol{\textlnot}{TS1}{172}

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

958 \DeclareTextSymbol{\textcircledP}{TS1}{173}
959 \DeclareTextSymbol{\textregistered}{TS1}{174}
960 \DeclareTextSymbol{\textasciimacron}{TS1}{175}

"B0 = 176
961 \DeclareTextSymbol{\textdegree}{TS1}{176}
962 \DeclareTextSymbol{\textpm}{TS1}{177}
963 \DeclareTextSymbol{\texttwosuperior}{TS1}{178}
964 \DeclareTextSymbol{\textthreesuperior}{TS1}{179}
965 \DeclareTextSymbol{\textasciacute}{TS1}{180}
966 \DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
967 \DeclareTextSymbol{\textparagraph}{TS1}{182}
968 \DeclareTextSymbol{\textperiodcentered}{TS1}{183}

"B8 = 184
969 \DeclareTextSymbol{\textreferencemark}{TS1}{184}
970 \DeclareTextSymbol{\textonesuperior}{TS1}{185}
971 \DeclareTextSymbol{\textordmasculine}{TS1}{186}
972 \DeclareTextSymbol{\textsurd}{TS1}{187}
973 \DeclareTextSymbol{\textonequarter}{TS1}{188}
974 \DeclareTextSymbol{\textonehalf}{TS1}{189}
975 \DeclareTextSymbol{\textthreequarters}{TS1}{190}
976 \DeclareTextSymbol{\texteuro}{TS1}{191}

"E0 = 208
977 \DeclareTextSymbol{\texttimes}{TS1}{214}

"F0 = 240
978 \DeclareTextSymbol{\textdiv}{TS1}{246}
979 </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 from the `inputenc` UTF-8 support, which is essentially doing a reverse mapping from UTF-8 data to L^AT_EX L^IC^R 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.

```
980 <*TU>
```

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

```
981 \providetcommand\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.

```
982 \begingroup\expandafter\expandafter\expandafter\endgroup
983 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
984   \begingroup\expandafter\expandafter\expandafter\endgroup
985   \expandafter\ifx\csname directlua\endcsname\relax
```

Not LuaTeX or XeTeX, abort with a warning.

```
986 \PackageWarningNoLine{fontenc}
987   {\UnicodeEncodingName\space
988     encoding is only available with XeTeX and LuaTeX.\MessageBreak
989     Defaulting to T1 encoding}
990   \def\encodingdefault{T1}
991   \expandafter\expandafter\expandafter\endinput
992 \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.

```
993 \def\UnicodeFontTeXLigatures{+tlig;}
994 \ifnum\luatexversion<110
995   \def\reserved@a#1{%
996     \def\@remove@tlig##1{\@remove@tlig##1\@nil#1\@nil\relax}
997     \def\@remove@tlig##1#1{\@remove@tlig##1#1}
998   \edef\reserved@b{\detokenize{+tlig;}}
999   \expandafter\reserved@a\expandafter{\reserved@b}
1000 \def\@remove@tlig##1\@nil#2\relax{#1}
```

```

1001      \def\remove@tlig#1{%
1002          \begingroup
1003          \font\remove@tlig
1004          \expandafter\@remove@tlig\expandafter{\fontname\font}%
1005          \remove@tlig
1006          \char#1\relax
1007          \endgroup
1008      }
1009  \else
1010      \newprotectedluacmd\@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.

```

1011      \now@and@everyjob{\directlua{
1012          local rawchar_func = token.create'@remove@tlig@@@'.index
1013          local forcehmode = tex.forcehmode
1014          local put_next = token.put_next
1015          local glyph_id = node.id'glyph'
1016          local rawchar_token = token.new(rawchar_func, token.command_id'lua_call')
1017          lua.get_functions_table()[rawchar_func] = function()
1018              local mode = tex.nest.top.mode
1019              if mode == 1 or mode == -1 then
1020                  put_next(rawchar_token)
1021                  return forcehmode(true)
1022              end
1023              local n = node.new(glyph_id, 256)
1024              n.font = font.current()
1025              n.char = token.scan_int()
1026              return node.write(n)
1027          end
1028      }}

```

Now `\remove@tlig` can be implemented almost as in XeTeX.

```

1029      \def\remove@tlig#1{\@remove@tlig@@@#1\relax}
1030      \fi
1031      \fi
1032  \else
XeTeX
1033  \def\UnicodeFontTeXLigatures{mapping=tex-text;}
1034  \def\remove@tlig#1{\XeTeXglyph\numexpr\XeTeXcharglyph#1\relax}
1035  \fi
1036 \def\UnicodeFontFile#1#2{"[#1]:#2"}
1037 \def\UnicodeFontName#1#2{"#1:#2"}

    Declare the encoding
1038 \DeclareFontEncoding\UnicodeEncodingName{}{}

    Declare accent command to use a postspended combining character rather than the
TeX \accent primitive
1039 \def\add@unicode@accent#1#2{%
1040   \if\relax\detokenize{#2}\relax^\wedge a0\else#2\fi
1041   \char#1\relax}

```

In its original implementation `\DeclareUnicodeAccent` was given 3 arguments (with second the “Unicode encoding” a.k.a., `\UnicodeEncodingName`) while in other places, e.g., `\DeclareUnicodeComposite`, we always made encoding implicit. So we now change it here to implicit too so that the interfaces become a bit more consistent. To avoid making that a breaking change (even though it only affects two packages on CTAN) we test for #2 being `\UnicodeEncodingName`. This would not catch if somebody used `\DeclareUnicodeAccent{\=}{TU-sub}{“0304”}` but that fortunately hasn’t happened. With the implicit argument you would need to change `\UnicodeEncodingName` instead, as you have to do anyway for the other interface commands.

```

1042 \def\DeclareUnicodeAccent#1#2{%
1043   \edef\reserved@a{#2}%
1044   \edef\reserved@b{\UnicodeEncodingName}%
1045   \ifx\reserved@a\reserved@b
1046     \def\reserved@a{\DeclareUnicodeAccent@{#1}}%
1047   \else
1048     \def\reserved@a{\DeclareUnicodeAccent@{#1}\UnicodeEncodingName}%
1049   \fi
1050   \reserved@a{#2}%
1051 }
1052 \def\DeclareUnicodeAccent@#1#2#3{%
1053   \DeclareTextCommand{#1}{#2}{\addunicode@accent{#3}}%
1054 }
```

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.

```

1055 {
1056   \catcode\z@=11\relax
1057   \gdef\DeclareUnicodeComposite#1#2#3{%
1058     \def\reserved@a##1##2{%
1059       \DeclareTextCompositeCommand#1\UnicodeEncodingName{#2}{%
1060         \iffontchar\font#3 ##2%
1061           \else ##1\fi}%
1062         \expandafter\expandafter\expandafter\extract@default@composite
1063         \csname\UnicodeEncodingName\string#1\endcsname{#2}\@nil
1064       \bgroup
1065         \lccode\z@#3 %
1066         \lowercase{\egroup
1067           \expandafter\reserved@a\expandafter{\reserved@b}{^{\z@}}}}%
1068     }
1069   \def\extract@default@composite#1{%
1070     \ifx\@text@composite#1%
1071       \expandafter\extract@default@composite@a
1072     \else
1073       \expandafter\extract@default@composite@b\expandafter#1%
1074     \fi}
1075   \def\extract@default@composite@a#1\@text@composite#2\@nil{%
1076     \def\reserved@b{#2}}
1077   \def\extract@default@composite@b#1#2\@nil{%
1078     \def\reserved@b{#1#2}}
```

Next two commands are simply syntactic sugar to go with the other `\DeclareUnicode...` declarations.

```

1079 \def\DeclareUnicodeSymbol#1{\DeclareTextSymbol{#1}{\UnicodeEncodingName}}
1080 \def\DeclareUnicodeCommand#1{\DeclareTextCommand{#1}{\UnicodeEncodingName}}
1081 \DeclareUnicodeCommand{textquotesingle} {\remove@tlig{"0027}}
1082 \DeclareUnicodeCommand{textasciigrave} {\remove@tlig{"0060}}
1083 \DeclareUnicodeCommand{textquotedbl} {\remove@tlig{"0022}}
1084 \DeclareUnicodeSymbol{textdollar} {"0024}
1085 \DeclareUnicodeSymbol{textless} {"003C}
1086 \DeclareUnicodeSymbol{textgreater} {"003E}
1087 \DeclareUnicodeSymbol{textbackslash} {"005C}
1088 \DeclareUnicodeSymbol{textasciicircum} {"005E}
1089 \DeclareUnicodeSymbol{textunderscore} {"005F}
1090 \DeclareUnicodeSymbol{textbraceleft} {"007B}
1091 \DeclareUnicodeSymbol{textbar} {"007C}
1092 \DeclareUnicodeSymbol{textbraceright} {"007D}
1093 \DeclareUnicodeSymbol{textasciitilde} {"007E}
1094 \DeclareUnicodeSymbol{textexclamdown} {"00A1}
1095 \DeclareUnicodeSymbol{textcent} {"00A2}
1096 \DeclareUnicodeSymbol{textsterling} {"00A3}
1097 \DeclareUnicodeSymbol{textcurrency} {"00A4}
1098 \DeclareUnicodeSymbol{textyen} {"00A5}
1099 \DeclareUnicodeSymbol{textbrokenbar} {"00A6}
1100 \DeclareUnicodeSymbol{textsection} {"00A7}
1101 \DeclareUnicodeSymbol{textasciidieresis} {"00A8}
1102 \DeclareUnicodeSymbol{textcopyright} {"00A9}
1103 \DeclareUnicodeSymbol{textordfeminine} {"00AA}

1104 \DeclareUnicodeSymbol{guillemetleft} {"00AB}
1105 % old Adobe name
1106 \DeclareUnicodeSymbol{guillemotleft} {"00AB}
1107 \DeclareUnicodeSymbol{textlnot} {"00AC}
1108 \DeclareUnicodeSymbol{textregistered} {"00AE}
1109 \DeclareUnicodeSymbol{textasciimacron} {"00AF}
1110 \DeclareUnicodeSymbol{textdegree} {"00B0}
1111 \DeclareUnicodeSymbol{textpm} {"00B1}
1112 \DeclareUnicodeSymbol{texttwosuperior} {"00B2}
1113 \DeclareUnicodeSymbol{textthreesuperior} {"00B3}
1114 \DeclareUnicodeSymbol{textasciacute} {"00B4}
1115 \DeclareUnicodeSymbol{textmu} {"00B5}
1116 \DeclareUnicodeSymbol{textparagraph} {"00B6}
1117 \DeclareUnicodeSymbol{textperiodcentered} {"00B7}
1118 \DeclareUnicodeSymbol{textonesuperior} {"00B9}
1119 \DeclareUnicodeSymbol{textordmasculine} {"00BA}

1120 \DeclareUnicodeSymbol{guillemetright} {"00BB}
1121 % old Adobe name
1122 \DeclareUnicodeSymbol{guillemotright} {"00BB}
1123 \DeclareUnicodeSymbol{textonequarter} {"00BC}
1124 \DeclareUnicodeSymbol{textonehalf} {"00BD}
1125 \DeclareUnicodeSymbol{textthreequarters} {"00BE}
1126 \DeclareUnicodeSymbol{textquestiondown} {"00BF}
1127 \DeclareUnicodeSymbol{AE} {"00C6}
1128 \DeclareUnicodeSymbol{DH} {"00D0}
1129 \DeclareUnicodeSymbol{texttimes} {"00D7}

```

```

1130 \DeclareUnicodeSymbol{\O} {"00D8}
1131 \DeclareUnicodeSymbol{\TH} {"00DE}
1132 \DeclareUnicodeSymbol{\ss} {"00DF}
1133 \DeclareUnicodeSymbol{\ae} {"00E6}
1134 \DeclareUnicodeSymbol{\dh} {"00F0}
1135 \DeclareUnicodeSymbol{\textdiv} {"00F7}
1136 \DeclareUnicodeSymbol{\o} {"00F8}
1137 \DeclareUnicodeSymbol{\th} {"00FE}
1138 \DeclareUnicodeSymbol{\DJ} {"0110}
1139 \DeclareUnicodeSymbol{\dj} {"0111}
1140 \DeclareUnicodeSymbol{\i} {"0131}
1141 \DeclareUnicodeSymbol{\IJ} {"0132}
1142 \DeclareUnicodeSymbol{\ij} {"0133}
1143 \DeclareUnicodeSymbol{\L} {"0141}
1144 \DeclareUnicodeSymbol{\l} {"0142}
1145 \DeclareUnicodeSymbol{\NG} {"014A}
1146 \DeclareUnicodeSymbol{\ng} {"014B}
1147 \DeclareUnicodeSymbol{\OE} {"0152}
1148 \DeclareUnicodeSymbol{\oe} {"0153}
1149 \DeclareUnicodeSymbol{\textflorin} {"0192}
1150 \DeclareUnicodeSymbol{\j} {"0237}
1151 \DeclareUnicodeSymbol{\textasciicaron} {"02C7}
1152 \DeclareUnicodeSymbol{\textasciibreve} {"02D8}
1153 \DeclareUnicodeSymbol{\textacutedbl} {"02DD}
1154 \DeclareUnicodeSymbol{\textgravedbl} {"02F5}
1155 \DeclareUnicodeSymbol{\texttildelow} {"02F7}
1156 \DeclareUnicodeSymbol{\textbaht} {"0E3F}
1157 \DeclareUnicodeSymbol{\SS} {"1E9E}
1158 \DeclareUnicodeSymbol{\textcompwordmark} {"200C}

1159 %\DeclareUnicodeSymbol{\textnonbreakinghyphen} {"2011}
1160 %\DeclareUnicodeSymbol{\textfiguredash} {"2012}
1161 \DeclareUnicodeSymbol{\textendash} {"2013}
1162 \DeclareUnicodeSymbol{\textemdash} {"2014}
1163 %\DeclareUnicodeSymbol{\texthorizontalbar} {"2015}

```

Unfortunately some fonts do not implement "2011, "2012 and/or "2015 (including the L^AT_EX default fonts for Unicode engines) so we provide some approximations if the glyph is missing, like we do for OT1 and T1.

The `\nobreak\hspace{z@}` is there to prevent a break after the hyphen but allow later breaks in the remainder of the word.

```

1164 \DeclareUnicodeCommand{\textnonbreakinghyphen}
1165   {\iffontchar\font "2011 \char "2011 \else \mbox{-}\nobreak\hspace{z@} \fi}
1166 \DeclareUnicodeCommand{\textfiguredash}
1167   {\iffontchar\font "2012 \char "2012 \else \char "2013 \fi}
1168 \DeclareUnicodeCommand{\texthorizontalbar}
1169   {\iffontchar\font "2015 \char "2015 \else \char "2014 \fi}

1170 \DeclareUnicodeSymbol{\textbardbl} {"2016}
1171 \DeclareUnicodeSymbol{\textquotel} {"2018}
1172 \DeclareUnicodeSymbol{\textquoter} {"2019}
1173 \DeclareUnicodeSymbol{\quotesinglbase} {"201A}
1174 \DeclareUnicodeSymbol{\textquotedblleft} {"201C}
1175 \DeclareUnicodeSymbol{\textquotedblright} {"201D}
1176 \DeclareUnicodeSymbol{\quotedblbase} {"201E}

```

```

1177 \DeclareUnicodeSymbol{\textdagger} {"2020}
1178 \DeclareUnicodeSymbol{\textdaggerdbl} {"2021}
1179 \DeclareUnicodeSymbol{\textbullet} {"2022}
1180 \DeclareUnicodeSymbol{\textellipsis} {"2026}
1181 \DeclareUnicodeSymbol{\textperthousand} {"2030}
1182 \DeclareUnicodeSymbol{\textpertenthousand} {"2031}
1183 \DeclareUnicodeSymbol{\guilsinglleft} {"2039}
1184 \DeclareUnicodeSymbol{\guilsinglright} {"203A}
1185 \DeclareUnicodeSymbol{\textreferencemark} {"203B}
1186 \DeclareUnicodeSymbol{\textinterrobang} {"203D}
1187 \DeclareUnicodeSymbol{\textfractionsolidus} {"2044}
1188 \DeclareUnicodeSymbol{\textlquill} {"2045}
1189 \DeclareUnicodeSymbol{\textrquill} {"2046}
1190 \DeclareUnicodeSymbol{\textdiscount} {"2052}
1191 \DeclareUnicodeSymbol{\textcolonmonetary} {"20A1}
1192 \DeclareUnicodeSymbol{\textlira} {"20A4}
1193 \DeclareUnicodeSymbol{\textnaira} {"20A6}
1194 \DeclareUnicodeSymbol{\textwon} {"20A9}
1195 \DeclareUnicodeSymbol{\textdong} {"20AB}
1196 \DeclareUnicodeSymbol{\texteuro} {"20AC}
1197 \DeclareUnicodeSymbol{\textpeso} {"20B1}
1198 \DeclareUnicodeSymbol{\textcelsius} {"2103}
1199 \DeclareUnicodeSymbol{\textnumero} {"2116}
1200 \DeclareUnicodeSymbol{\textcircledP} {"2117}
1201 \DeclareUnicodeSymbol{\textrecipie} {"211E}
1202 \DeclareUnicodeSymbol{\textservicemark} {"2120}
1203 \DeclareUnicodeSymbol{\texttrademark} {"2122}
1204 \DeclareUnicodeSymbol{\textohm} {"2126}
1205 \DeclareUnicodeSymbol{\textmho} {"2127}
1206 \DeclareUnicodeSymbol{\textestimated} {"212E}
1207 \DeclareUnicodeSymbol{\textleftarrow} {"2190}
1208 \DeclareUnicodeSymbol{\textuparrow} {"2191}
1209 \DeclareUnicodeSymbol{\textrightarrow} {"2192}
1210 \DeclareUnicodeSymbol{\textdownarrow} {"2193}
1211 \DeclareUnicodeSymbol{\textminus} {"2212}
1212
1213 \DeclareUnicodeSymbol{\Hwithstroke} {"0126}
1214 \DeclareUnicodeSymbol{\hwithstroke} {"0127}

```

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

```

1215 \DeclareUnicodeCommand{\textasteriskcentered}{%
1216   \iffontchar\font"2217 \char"2217 \else
1217     \begingroup
1218       \fontsize
1219         {\the\dimexpr1.3\dimexpr\f@size pt\relax}%
1220         {\f@baselineskip}%
1221       \selectfont
1222         \raisebox{-0.7ex}{[\dimexpr\height-0.7ex] [0pt]{*}}%
1223     \endgroup
1224   \fi
1225 }
1226 \DeclareUnicodeSymbol{\textsurd} {"221A}
1227 \DeclareUnicodeSymbol{\textlangle} {"2329}

```

```

1228 \DeclareUnicodeSymbol{\textrangle} {"232A}
1229 \DeclareUnicodeSymbol{\textblank} {"2422}
1230 \DeclareUnicodeSymbol{\textvisiblespace} {"2423}
1231 \DeclareUnicodeSymbol{\textopenbullet} {"25E6}
1232 \DeclareUnicodeSymbol{\textbigcircle} {"25EF}
1233 \DeclareUnicodeSymbol{\textmusicalnote} {"266A}
1234 \DeclareUnicodeSymbol{\textmarried} {"26AD}
1235 \DeclareUnicodeSymbol{\textdivorced} {"26AE}
1236 \DeclareUnicodeSymbol{\textinterrobangdown} {"2E18}

```

Accents must be declared before the composites that use them.

```

1237 \DeclareUnicodeAccent{\`}{0300}
1238 \DeclareUnicodeAccent{\'}{0301}
1239 \DeclareUnicodeAccent{\^}{0302}
1240 \DeclareUnicodeAccent{\~}{0303}
1241 \DeclareUnicodeAccent{\=}{0304}
1242 \DeclareUnicodeAccent{\u}{0306}
1243 \DeclareUnicodeAccent{\.}{0307}
1244 \DeclareUnicodeAccent{\"}{0308}
1245 \DeclareUnicodeAccent{\r}{030A}
1246 \DeclareUnicodeAccent{\H}{030B}
1247 \DeclareUnicodeAccent{\v}{030C}
1248 \DeclareUnicodeAccent{\b}{0332}
1249 \DeclareUnicodeAccent{\d}{0323}
1250 \DeclareUnicodeAccent{\c}{0327}
1251 \DeclareUnicodeAccent{\k}{0328}

```

The odd one out:

```

1252 \DeclareUnicodeCommand{textcommabelow[1]
1253   {\hmode@bgroup\oalign{\null#1\crcr\hidewidth\raise-.31ex
1254     \hbox{\check@mathfonts\fontsize\ssf@size\z@
1255       \math@fontsfalse\selectfont,\}\hidewidth}\egroup}
1256 \DeclareUnicodeComposite{\^} {"005E}
1257 \DeclareUnicodeComposite{\~} {"007E}
1258 \DeclareUnicodeComposite{\`} {"00C0}
1259 \DeclareUnicodeComposite{\'} {"00C1}
1260 \DeclareUnicodeComposite{\^} {"00C2}
1261 \DeclareUnicodeComposite{\~} {"00C3}
1262 \DeclareUnicodeComposite{\"} {"00C4}
1263 \DeclareUnicodeComposite{\r} {"00C5}
1264 \DeclareUnicodeComposite{\c} {"00C7}
1265 \DeclareUnicodeComposite{\`} {"00C8}
1266 \DeclareUnicodeComposite{\'} {"00C9}
1267 \DeclareUnicodeComposite{\^} {"00CA}
1268 \DeclareUnicodeComposite{\"} {"00CB}
1269 \DeclareUnicodeComposite{\`} {"00CC}
1270 \DeclareUnicodeComposite{\'} {"00CD}
1271 \DeclareUnicodeComposite{\^} {"00CE}
1272 \DeclareUnicodeComposite{\"} {"00CF}
1273 \DeclareUnicodeComposite{\~} {"00D1}
1274 \DeclareUnicodeComposite{\`} {"00D2}
1275 \DeclareUnicodeComposite{\'} {"00D3}
1276 \DeclareUnicodeComposite{\^} {"00D4}
1277 \DeclareUnicodeComposite{\~} {"00D5}

```

```

1278 \DeclareUnicodeComposite{\"} {0}{"00D6}
1279 \DeclareUnicodeComposite{\'} {U}{"00D9}
1280 \DeclareUnicodeComposite{\'} {U}{"00DA}
1281 \DeclareUnicodeComposite{\^} {U}{"00DB}
1282 \DeclareUnicodeComposite{\"} {U}{"00DC}
1283 \DeclareUnicodeComposite{\'} {Y}{"00DD}
1284 \DeclareUnicodeComposite{\'} {a}{"00E0}
1285 \DeclareUnicodeComposite{\'} {a}{"00E1}
1286 \DeclareUnicodeComposite{\^} {a}{"00E2}
1287 \DeclareUnicodeComposite{\~} {a}{"00E3}
1288 \DeclareUnicodeComposite{\"} {a}{"00E4}
1289 \DeclareUnicodeComposite{\r} {a}{"00E5}
1290 \DeclareUnicodeComposite{\c} {c}{"00E7}
1291 \DeclareUnicodeComposite{\'} {e}{"00E8}
1292 \DeclareUnicodeComposite{\'} {e}{"00E9}
1293 \DeclareUnicodeComposite{\^} {e}{"00EA}
1294 \DeclareUnicodeComposite{\"} {e}{"00EB}
1295 \DeclareUnicodeComposite{\'} {\i {"00EC}}
1296 \DeclareUnicodeComposite{\'} {\i {"00EC}}
1297 \DeclareUnicodeComposite{\'} {\i {"00ED}}
1298 \DeclareUnicodeComposite{\'} {\i {"00ED}}
1299 \DeclareUnicodeComposite{\^} {\i {"00EE}}
1300 \DeclareUnicodeComposite{\~} {\i {"00EE}}
1301 \DeclareUnicodeComposite{\"} {\i {"00EF}}
1302 \DeclareUnicodeComposite{\\"} {\i {"00EF}}
1303 \DeclareUnicodeComposite{\~} {\n {"00F1}}
1304 \DeclareUnicodeComposite{\'} {\o {"00F2}}
1305 \DeclareUnicodeComposite{\'} {\o {"00F3}}
1306 \DeclareUnicodeComposite{\^} {\o {"00F4}}
1307 \DeclareUnicodeComposite{\~} {\o {"00F5}}
1308 \DeclareUnicodeComposite{\"} {\o {"00F6}}
1309 \DeclareUnicodeComposite{\'} {\u {"00F9}}
1310 \DeclareUnicodeComposite{\'} {\u {"00FA}}
1311 \DeclareUnicodeComposite{\^} {\u {"00FB}}
1312 \DeclareUnicodeComposite{\\"} {\u {"00FC}}
1313 \DeclareUnicodeComposite{\'} {\y {"00FD}}
1314 \DeclareUnicodeComposite{\\"} {\y {"00FF}}
1315 \DeclareUnicodeComposite{\=} {\A {"0100}}
1316 \DeclareUnicodeComposite{\=} {\a {"0101}}
1317 \DeclareUnicodeComposite{\u} {\A {"0102}}
1318 \DeclareUnicodeComposite{\u} {\a {"0103}}
1319 \DeclareUnicodeComposite{\k} {\A {"0104}}
1320 \DeclareUnicodeComposite{\k} {\a {"0105}}
1321 \DeclareUnicodeComposite{\'} {\C {"0106}}
1322 \DeclareUnicodeComposite{\'} {\c {"0107}}
1323 \DeclareUnicodeComposite{\~} {\C {"0108}}
1324 \DeclareUnicodeComposite{\~} {\c {"0109}}
1325 \DeclareUnicodeComposite{\.} {\C {"010A}}
1326 \DeclareUnicodeComposite{\.} {\c {"010B}}
1327 \DeclareUnicodeComposite{\v} {\C {"010C}}
1328 \DeclareUnicodeComposite{\v} {\c {"010D}}
1329 \DeclareUnicodeComposite{\v} {\D {"010E}}
1330 \DeclareUnicodeComposite{\v} {\d {"010F}}
1331 \DeclareUnicodeComposite{\=} {\E {"0112}}

```

```

1332 \DeclareUnicodeComposite{\=}{e}{0113}
1333 \DeclareUnicodeComposite{\u}{E}{0114}
1334 \DeclareUnicodeComposite{\u}{e}{0115}
1335 \DeclareUnicodeComposite{\.}{E}{0116}
1336 \DeclareUnicodeComposite{\.}{e}{0117}
1337 \DeclareUnicodeComposite{\k}{E}{0118}
1338 \DeclareUnicodeComposite{\k}{e}{0119}
1339 \DeclareUnicodeComposite{\v}{E}{011A}
1340 \DeclareUnicodeComposite{\v}{e}{011B}
1341 \DeclareUnicodeComposite{\^}{G}{011C}
1342 \DeclareUnicodeComposite{\^}{g}{011D}
1343 \DeclareUnicodeComposite{\u}{G}{011E}
1344 \DeclareUnicodeComposite{\u}{g}{011F}
1345 \DeclareUnicodeComposite{\.}{G}{0120}
1346 \DeclareUnicodeComposite{\.}{g}{0121}
1347 \DeclareUnicodeComposite{\c}{G}{0122}
1348 \DeclareUnicodeComposite{\c}{g}{0123}
1349 \DeclareUnicodeComposite{\~}{H}{0124}
1350 \DeclareUnicodeComposite{\~}{h}{0125}
1351 \DeclareUnicodeComposite{\~}{I}{0128}
1352 \DeclareUnicodeComposite{\~}{i}{0129}
1353 \DeclareUnicodeComposite{\~}{i}{0129}
1354 \DeclareUnicodeComposite{\=}{I}{012A}
1355 \DeclareUnicodeComposite{\=}{i}{012B}
1356 \DeclareUnicodeComposite{\=}{i}{012B}
1357 \DeclareUnicodeComposite{\u}{I}{012C}
1358 \DeclareUnicodeComposite{\u}{i}{012D}
1359 \DeclareUnicodeComposite{\u}{i}{012D}
1360 \DeclareUnicodeComposite{\k}{I}{012E}
1361 \DeclareUnicodeComposite{\k}{i}{012F}
1362 \DeclareUnicodeComposite{\k}{i}{012F}
1363 \DeclareUnicodeComposite{\.}{I}{0130}
1364 \DeclareUnicodeComposite{\^}{J}{0134}
1365 \DeclareUnicodeComposite{\^}{j}{0135}
1366 \DeclareUnicodeComposite{\^}{j}{0135}
1367 \DeclareUnicodeComposite{\c}{K}{0136}
1368 \DeclareUnicodeComposite{\c}{k}{0137}
1369 \DeclareUnicodeComposite{\'}{L}{0139}
1370 \DeclareUnicodeComposite{\'}{l}{013A}
1371 \DeclareUnicodeComposite{\c}{L}{013B}
1372 \DeclareUnicodeComposite{\c}{l}{013C}
1373 \DeclareUnicodeComposite{\v}{L}{013D}
1374 \DeclareUnicodeComposite{\v}{l}{013E}
1375 \DeclareUnicodeComposite{\'}{N}{0143}
1376 \DeclareUnicodeComposite{\'}{n}{0144}
1377 \DeclareUnicodeComposite{\c}{N}{0145}
1378 \DeclareUnicodeComposite{\c}{n}{0146}
1379 \DeclareUnicodeComposite{\v}{N}{0147}
1380 \DeclareUnicodeComposite{\v}{n}{0148}
1381 \DeclareUnicodeComposite{\=}{O}{014C}
1382 \DeclareUnicodeComposite{\=}{o}{014D}
1383 \DeclareUnicodeComposite{\u}{O}{014E}
1384 \DeclareUnicodeComposite{\u}{o}{014F}
1385 \DeclareUnicodeComposite{\H}{O}{0150}

```

```

1386 \DeclareUnicodeComposite{\H} {o}{0151}
1387 \DeclareUnicodeComposite{\'} {R}{0154}
1388 \DeclareUnicodeComposite{\'} {r}{0155}
1389 \DeclareUnicodeComposite{\c} {R}{0156}
1390 \DeclareUnicodeComposite{\c} {r}{0157}
1391 \DeclareUnicodeComposite{\v} {R}{0158}
1392 \DeclareUnicodeComposite{\v} {r}{0159}
1393 \DeclareUnicodeComposite{\'} {S}{015A}
1394 \DeclareUnicodeComposite{\'} {s}{015B}
1395 \DeclareUnicodeComposite{\^} {S}{015C}
1396 \DeclareUnicodeComposite{\^} {s}{015D}
1397 \DeclareUnicodeComposite{\c} {S}{015E}
1398 \DeclareUnicodeComposite{\c} {s}{015F}
1399 \DeclareUnicodeComposite{\v} {S}{0160}
1400 \DeclareUnicodeComposite{\v} {s}{0161}
1401 \DeclareUnicodeComposite{\c} {T}{0162}
1402 \DeclareUnicodeComposite{\c} {t}{0163}
1403 \DeclareUnicodeComposite{\v} {T}{0164}
1404 \DeclareUnicodeComposite{\v} {t}{0165}
1405 \DeclareUnicodeComposite{\~} {U}{0168}
1406 \DeclareUnicodeComposite{\~} {u}{0169}
1407 \DeclareUnicodeComposite{\=} {U}{016A}
1408 \DeclareUnicodeComposite{\=} {u}{016B}
1409 \DeclareUnicodeComposite{\u} {U}{016C}
1410 \DeclareUnicodeComposite{\u} {u}{016D}
1411 \DeclareUnicodeComposite{\r} {U}{016E}
1412 \DeclareUnicodeComposite{\r} {u}{016F}
1413 \DeclareUnicodeComposite{\H} {U}{0170}
1414 \DeclareUnicodeComposite{\H} {u}{0171}
1415 \DeclareUnicodeComposite{\k} {U}{0172}
1416 \DeclareUnicodeComposite{\k} {u}{0173}
1417 \DeclareUnicodeComposite{\^} {W}{0174}
1418 \DeclareUnicodeComposite{\^} {w}{0175}
1419 \DeclareUnicodeComposite{\~} {Y}{0176}
1420 \DeclareUnicodeComposite{\~} {y}{0177}
1421 \DeclareUnicodeComposite{\"} {Y}{0178}
1422 \DeclareUnicodeComposite{\'} {Z}{0179}
1423 \DeclareUnicodeComposite{\'} {z}{017A}
1424 \DeclareUnicodeComposite{\.} {Z}{017B}
1425 \DeclareUnicodeComposite{\.} {z}{017C}
1426 \DeclareUnicodeComposite{\v} {Z}{017D}
1427 \DeclareUnicodeComposite{\v} {z}{017E}
1428 \DeclareUnicodeComposite{\v} {A}{01CD}
1429 \DeclareUnicodeComposite{\v} {a}{01CE}
1430 \DeclareUnicodeComposite{\v} {I}{01CF}
1431 \DeclareUnicodeComposite{\v} {i}{01D0}
1432 \DeclareUnicodeComposite{\v} {i}{01D0}
1433 \DeclareUnicodeComposite{\v} {O}{01D1}
1434 \DeclareUnicodeComposite{\v} {o}{01D2}
1435 \DeclareUnicodeComposite{\v} {U}{01D3}
1436 \DeclareUnicodeComposite{\v} {u}{01D4}

1437 \DeclareUnicodeComposite{\'} {\AE}{01FC}
1438 \DeclareUnicodeComposite{\'} {\xE}{01FC}

```

```

1439 \DeclareUnicodeComposite{\'}          \ae{"01FD}
1440 \DeclareUnicodeComposite{\'}          {\æ}{"01FD}
1441 \DeclareUnicodeComposite{\=}          \AE{"01E2}
1442 \DeclareUnicodeComposite{\=}          {\Œ}{"01E2}
1443 \DeclareUnicodeComposite{\=}          \ae{"01E3}
1444 \DeclareUnicodeComposite{\=}          {\æ}{"01E3}
1445 \DeclareUnicodeComposite{\v}          \G{"01E6}
1446 \DeclareUnicodeComposite{\v}          \g{"01E7}
1447 \DeclareUnicodeComposite{\v}          \K{"01E8}
1448 \DeclareUnicodeComposite{\v}          \k{"01E9}
1449 \DeclareUnicodeComposite{\k}          \O{"01EA}
1450 \DeclareUnicodeComposite{\k}          \o{"01EB}
1451 \DeclareUnicodeComposite{\v}          \j {"01FO}
1452 \DeclareUnicodeComposite{\v}          \j {"01FO}
1453 \DeclareUnicodeComposite{\'}          \G{"01F4}
1454 \DeclareUnicodeComposite{\'}          \g{"01F5}
1455 \DeclareUnicodeComposite{\textcommabelow}{S}{"0218}
1456 \DeclareUnicodeComposite{\textcommabelow}{s}{"0219}
1457 \DeclareUnicodeComposite{\textcommabelow}{T}{"021A}
1458 \DeclareUnicodeComposite{\textcommabelow}{t}{"021B}
1459 \DeclareUnicodeComposite{\=}          \Y{"0232}
1460 \DeclareUnicodeComposite{\=}          \y {"0233}
1461 \DeclareUnicodeComposite{\.}          \B {"1E02}
1462 \DeclareUnicodeComposite{\.}          \b {"1E03}
1463 \DeclareUnicodeComposite{\d}          \B {"1E04}
1464 \DeclareUnicodeComposite{\d}          \b {"1E05}
1465 \DeclareUnicodeComposite{\d}          \D {"1E0C}
1466 \DeclareUnicodeComposite{\d}          \d {"1E0D}
1467 \DeclareUnicodeComposite{\=}          \G {"1E20}
1468 \DeclareUnicodeComposite{\=}          \g {"1E21}
1469 \DeclareUnicodeComposite{\d}          \H {"1E24}
1470 \DeclareUnicodeComposite{\d}          \h {"1E25}
1471 \DeclareUnicodeComposite{\d}          \K {"1E32}
1472 \DeclareUnicodeComposite{\d}          \k {"1E33}
1473 \DeclareUnicodeComposite{\d}          \L {"1E36}
1474 \DeclareUnicodeComposite{\d}          \l {"1E37}
1475 \DeclareUnicodeComposite{\d}          \M {"1E42}
1476 \DeclareUnicodeComposite{\d}          \m {"1E43}
1477 \DeclareUnicodeComposite{\d}          \N {"1E46}
1478 \DeclareUnicodeComposite{\d}          \n {"1E47}
1479 \DeclareUnicodeComposite{\d}          \R {"1E5A}
1480 \DeclareUnicodeComposite{\d}          \r {"1E5B}
1481 \DeclareUnicodeComposite{\d}          \S {"1E62}
1482 \DeclareUnicodeComposite{\d}          \s {"1E63}
1483 \DeclareUnicodeComposite{\d}          \T {"1E6C}
1484 \DeclareUnicodeComposite{\d}          \t {"1E6D}
1485 \DeclareUnicodeComposite{\d}          \V {"1E7E}
1486 \DeclareUnicodeComposite{\d}          \v {"1E7F}
1487 \DeclareUnicodeComposite{\d}          \W {"1E88}
1488 \DeclareUnicodeComposite{\d}          \w {"1E89}
1489 \DeclareUnicodeComposite{\d}          \Z {"1E92}
1490 \DeclareUnicodeComposite{\d}          \z {"1E93}
1491 \DeclareUnicodeComposite{\d}          \A {"1EA0}
1492 \DeclareUnicodeComposite{\d}          \a {"1EA1}

```

```

1493 \DeclareUnicodeComposite{\d} {E}{1EB8}
1494 \DeclareUnicodeComposite{\d} {e}{1EB9}
1495 \DeclareUnicodeComposite{\d} {I}{1ECA}
1496 \DeclareUnicodeComposite{\d} {i}{1ECB}
1497 \DeclareUnicodeComposite{\d} {O}{1ECC}
1498 \DeclareUnicodeComposite{\d} {o}{1ECD}
1499 \DeclareUnicodeComposite{\d} {U}{1EE4}
1500 \DeclareUnicodeComposite{\d} {u}{1EE5}
1501 \DeclareUnicodeComposite{\d} {Y}{1EF4}
1502 \DeclareUnicodeComposite{\d} {y}{1EF5}

1503 
```

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.

```
1504 <*package>
```

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

```

1505 \def\update@uclc@with@cyrilllic{%
1506   \expandafter\def\expandafter@\uclclist\expandafter
1507   {\uclclist
1508     \cyra\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
1509     \CYRABHDZE\cyrabhha\CYRABHHA\cyrae\CYRAE\cyrb\CYRB\cyrbyus
1510     \CYRBYUS\cycrc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrdsc
1511     \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
1512     \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
1513     \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
1514     \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
1515     \cyrgchcrs\CYRGHCRS\cyrghk\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
1516     \cyrhdsc\CYRHDSC\cyrhhcrs\CYRHHCRS\cyrrhk\CYRHHK\cyrrhdsn
1517     \CYRHRDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
1518     \cyrishrtsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
1519     \cyrkbeak\CYRKBEAK\cyrkdsc\CYRKDSC\cyrkhcrs\CYRKHCRS\cyrkhk
1520     \CYRKHK\cyrkvcrs\CYRKVCRS\cyrl\CYRL\cyrlldsc\CYRLDSC\cyrlhk
1521     \CYRLHK\cyrlje\CYRLJE\cyrm\CYRM\cyrmndsc\CYRMNDSC\cyrmhk\CYRMHK
1522     \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnhk\CYRNHK\cyrnje
1523     \CYRNJE\cyrnlhk\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
1524     \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
1525     \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
1526     \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN

```

```

1527   \cyrstfn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
1528   \cyrt\CYRT\cyrt\cyrtdsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
1529   \cyr\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\ciry\CYRY
1530   \carya\CYRYA\caryat\CYRYAT\caryhcrs\CYRYHCRS\caryi\CYRYI\caryo
1531   \CYRYO\caryu\CYRYU\caryz\CYRZ\carydsc\CYRZDSC\caryzh\CYRZH
1532   \caryhdsc\CYRZHDSC}%
1533 \let\update@uclc@with@cyrillic\relax
1534 }

```

Here we process each option:

```

1535 \DeclareOption*{%
1536   \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 at whether `\T@encoding` is already defined. If not, we load it later (indicated by setting the switch `@tempswa` to true) and we always load if we are using an older format (or rather in a rollback situation).

```

1537   \tempswafalse
1538   \ifl@t@r\fmtversion{2020/02/02}%
1539     {\expandafter\ifx\csname T@\CurrentOption\endcsname\relax
1540       \tempswatrue\fi}%
1541     {\tempswatrue}%

```

Load if necessary:

```

1542 \if@tempswa
1543   \edef\reserved@f{%
1544     \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
1545   \reserved@f
1546   \InputIfFileExists\reserved@f
1547     {}{\PackageError{fontenc}%
1548       {Encoding file '\reserved@f' not found.%%
1549        \MessageBreak
1550        You might have misspelled the name of the encoding%
1551        \MessageBreak
1552        or a required support package (e.g., cyrillic) is%
1553        \MessageBreak
1554        missing in your installation}%
1555       {Necessary code for this encoding was not%
1556        loaded.\MessageBreak
1557        Thus calling the encoding later on will%
1558        produce further error messages.}}%
1559   \let\reserved@f\relax

```

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

```

1560   \expandafter\in@\expandafter{\CurrentOption}%
1561           {T2A,T2B,T2C,X2,LCY,OT2}%
1562   \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.

```

1563   \expandafter\in@\expandafter\cyla\expandafter
1564           {\@uclclist}%
1565   \ifin@

```

```

1566     \else
1567         \update@uclc@with@cyrillic
1568     \fi
1569     \fi
1570 \fi
1571 }
1572 \ProcessOptions*

```

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

```
1573 \%fontencoding\encodingdefault\selectfont
```

So instead we do this here:

```
1574 \usefont\encodingdefault\familydefault\seriesdefault\shapedefault
```

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

```
1575 \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.

```

1576 \let\@elt\relax
1577 \xdef\@fontenc@load@list{\@fontenc@load@list
1578   \@elt{\csname opt@fontenc.sty\endcsname}}
1579 \global\expandafter\let\csname ver@@fontenc.sty\expandafter\endcsname
1580   \csname ver@fontenc.sty\endcsname
1581 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
1582 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
1583 \global\let\@ifl@ter@\@ifl@ter
1584 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
1585 
```

File 22

ltcounts.dtx

1 Counters and Lengths

Commands for defining and using counters. This file defines:

| | |
|------------------------------|--|
| <code>\newcounter</code> | To define a new counter. |
| <code>\setcounter</code> | To set the value of counters. |
| <code>\addtocounter</code> | Increase the counter #1 by the number #2. |
| <code>\stepcounter</code> | Increase a counter by one. |
| <code>\refstepcounter</code> | Increase a counter by one, also setting the value used by <code>\label</code> . |
| <code>\value</code> | For accessing the value of the counter as a T _E X number (as opposed to <code>\the<counter></code> which expands to the <i>printed</i> representation of <code><counter></code>) |
| <code>\arabic</code> | <code>\arabic{<counter>}</code> : 1, 2, 3, ... |
| <code>\roman</code> | <code>\roman{<counter>}</code> : i, ii, iii, ... |
| <code>\Roman</code> | <code>\Roman{<counter>}</code> : I, II, III, ... |
| <code>\alph</code> | <code>\alph{<counter>}</code> : a, b, c, ... |
| <code>\Alpha</code> | <code>\Alpha{<counter>}</code> : A, B, C, ... |
| <code>\fnsymbol</code> | <code>\fnsymbol{<counter>}</code> : *, †, ‡, ... |
| <code>\counterwithin</code> | <code>\counterwithin[<format>]{<counter>}{<within-counter>}</code> : Resets <code><counter></code> whenever <code><within-counter></code> is stepped. Also redefines <code>\the<counter></code> command to produce <code>\the<within-counter>.<format>{<counter>}</code> with <code>\arabic</code> as the default for <code><format></code> . Star form omits redefining the print representation. |
| <code>\counterwithout</code> | <code>\counterwithout[<format>]{<counter>}{<within-counter>}</code> : Removes <code><counter></code> from the reset list of <code><within-counter></code> . Also redefines <code>\the<counter></code> command to produce <code><format>{<counter>}</code> with <code>\arabic</code> as the default for <code><format></code> . Star form omits redefining the print representation. |

1 (*2ekernel)

1.1 Environment Counter Macros

An environment foo has an associated counter defined by the following control sequences:

| | |
|----------------------|--|
| <code>\c@foo</code> | Contains the counter's numerical value. It is defined by <code>\newcount\foocounter</code> . |
| <code>\thefoo</code> | Macro that expands to the printed value of <code>\foocounter</code> . For example, if sections are numbered within chapters, and section headings look like |
| | Section II-3. The Nature of Counters |
| | then <code>\thesection</code> might be defined by: |
| | <code>\def\thesection</code> <code>{\@Roman{\c@chapter}-\@arabic{\c@section}}</code> |
| <code>\p@foo</code> | Macro that expands to a printed 'reference prefix' of counter foo. Any <code>\ref</code> to a value created by counter foo will produce the expansion of <code>\p@foo\thefoo</code> when the <code>\label</code> command is executed. See file <code>ltxref.dtx</code> for an extension of this mechanism. |
| <code>\cl@foo</code> | List of counters to be reset when foo stepped. Has format <code>\@elt{counterA}\@elt{counterB}\@elt{counterC}</code> . |

NOTE:

`\thefoo` and `\p@foo` must be defined in such a way that `\edef\bar{\thefoo}` or `\edef\bar{\p@foo}` defines `\bar` so that it will evaluate to the counter value at the time of the `\edef`, even after `\foocounter` and any other counters have been changed. This will happen if you use the standard commands `\@arabic`, `\@Roman`, etc.

The following commands are used to define and modify counters.

`\refstepcounter{<foo>}`

Same as `\stepcounter`, but it also defines `\@currentlabel`, `\@currentHref` and `\@currentcounter` and 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 and `\theHfoo` to be `\number\value{foo}`. 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{<foo>}{<val>}` : Globally sets `\foocounter` equal to `<val>`.

```
2 \def\setcounter#1#2{%
3   \@ifundefined{c@#1}%
4     {\@nocounterr{#1}}%
5     {\global\csname c@#1\endcsname#2\relax}}
```

(End of definition for `\setcounter`.)

`\addtocounter{<foo>}{<val>}` Globally increments `\foocounter` by `<val>`.

```
6 \def\addtocounter#1#2{%
7   \@ifundefined{c@#1}%
8     {\@nocounterr{#1}}%
9     {\global\advance\csname c@#1\endcsname #2\relax}}
```

(End of definition for `\addtocounter`.)

`\newcounter{<newctr>}[<oldctr>]` Defines `<newctr>` to be a counter, which is reset when counter `<oldctr>` is stepped. If `<newctr>` already defined produces ‘`c@newctr already defined`’ error.

```
10 \def\newcounter#1{%
11   \expandafter\@if definable \csname c@#1\endcsname
12   {\@definecounter{#1}}%
13   \@ifnextchar[\{\@newctr{#1}\}{}]
```

(End of definition for `\newcounter`.)

`\value{<ctr>}` produces the value of counter `<ctr>`, for use with a `\setcounter` or `\addtocounter` command.

```
14 \def\value#1{\csname c@#1\endcsname}
```

(End of definition for `\value`.)

`\@newctr`

```
15 \def\@newctr#1[#2]{%
16   \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}
```

(End of definition for \@newctr.)

\stepcounter \stepcounterfoo Globally increments counter \c@FOO and resets all subsidiary counters.

```
17 \def\stepcounter#1{%
18   \addtocounter{#1}\@ne
19   \begingroup
20     \let\@elt\@stpelt
21     \csname cl@#1\endcsname
22   \endgroup}
```

(End of definition for \stepcounter.)

\@stpelt Rather than resetting the “within” counter to zero we set it to -1 and then run \stepcounter that moves it to 0 and also initiates resetting the next level down.

```
23 </2ekernel>
24 <latexrelease>\IncludeInRelease[2015/01/01]{\@stpelt}
25 <latexrelease>                                {Reset nested counters}%
26 <*2ekernel | latexrelease>
27 \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne\stepcounter{#1}}%
28 <latexrelease>\EndIncludeInRelease
29 </2ekernel | latexrelease>
30 <latexrelease>\IncludeInRelease[0000/00/00]{\@stpelt}
31 <latexrelease>                                {Reset nested counters}%%
32 <latexrelease>\def\@stpelt#1{\global\csname c@#1\endcsname \z@}%
33 <latexrelease>\EndIncludeInRelease
34 <*2ekernel>
```

(End of definition for \@stpelt.)

\cl@ckpt

```
35 \def\cl@ckpt{\@elt{page}}
```

(End of definition for \cl@ckpt.)

\@definecounter

```
36 % \changes{v1.1p}{2024/10/26}{Fully expand counter name in theHfoo commands (gh/1508)}
37 </2ekernel>
38 <latexrelease>\IncludeInRelease[2024/11/01]{\@definecounter}
39 <latexrelease>                                {provide theHfoo commands}%
40 <*2ekernel | latexrelease>
41 \def\@definecounter#1{\expandafter\newcount\csname c@#1\endcsname
42   \setcounter{#1}\z@
43   \global\expandafter\let\csname cl@#1\endcsname\@empty
44   \@addtoreset{#1}{\@ckpt}%
45   \global\expandafter\let\csname p@#1\endcsname\@empty
46   \expandafter\xdef\csname theH#1\endcsname{%
47     \noexpand\the\noexpand\value{#1}}%
```

If \the#1 is undefined or \relax we define it with the standard definition for counters, otherwise we warn. This will catch, for example, that somebody defines a counter named “index” conflicting with the theindex environment.

```
48   \expandafter
49   \ifx\csname the#1\endcsname\relax
50     \expandafter
51     \gdef\csname the#1\expandafter\endcsname\expandafter
```

```

52          {\expandafter\@arabic\csname c@\#1\endcsname}%
53 \else
54     \@latex@warning{Command '\string\the#1' already
55                     defined -- not changed}%
56 \fi}
57 <|latexrelease>\EndIncludeInRelease
58 </2ekernel | latexrelease>
59 <|latexrelease>\IncludeInRelease{0000/00/00}{\@definecounter}
60 <|latexrelease>                                         {provide theHfoo commands}%%
61 <|latexrelease>\def\@definecounter#1{\expandafter\newcount\csname c@\#1\endcsname
62 <|latexrelease>      \setcounter{#1}\z@
63 <|latexrelease>      \global\expandafter\let\csname cl@#1\endcsname\@empty
64 <|latexrelease>      \addtoreset{#1}{\@ckpt}%
65 <|latexrelease>      \global\expandafter\let\csname p@#1\endcsname\@empty
66 <|latexrelease>      \expandafter
67 <|latexrelease>      \ifx\csname the#1\endcsname\relax
68 <|latexrelease>          \expandafter
69 <|latexrelease>          \gdef\csname the#1\expandafter\endcsname\expandafter
70 <|latexrelease>              {\expandafter\@arabic\csname c@\#1\endcsname}%
71 <|latexrelease>          \else
72 <|latexrelease>              \@latex@warning{Command '\string\the#1' already
73 <|latexrelease>                      defined -- not changed}%
74 <|latexrelease>          \fi}
75 <|latexrelease>\EndIncludeInRelease
76 <*2ekernel>

```

(End of definition for \@definecounter.)

\@addtoreset If a counter is reset when a parent counter changes it no longer has an unique value across the document. As \theH<counter> should be unique this representation is changed to include also the representation of the parent. This is not 100% guaranteed to work but has been used this way by hyperref for many years without causing problems.

```

77 </2ekernel>
78 <|latexrelease>\IncludeInRelease{2024/11/01}{\@addtoreset}
79 <|latexrelease>                                         {provide theHfoo commands}%%
80 <*2ekernel | latexrelease>
81 \def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}%
82   \expandafter\xdef\csname theH#1\endcsname{%
83     \expandafter\noexpand\csname theH#2\endcsname.%%
84     \noexpand\the\noexpand\value{#1}}%
85 }
86 <|latexrelease>\EndIncludeInRelease
87 </2ekernel | latexrelease>
88 <|latexrelease>\IncludeInRelease{0000/00/00}{\@addtoreset}
89 <|latexrelease>                                         {provide theHfoo commands}%%
90 <|latexrelease>\def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}}
91 <|latexrelease>\EndIncludeInRelease
92 <*2ekernel>

```

(End of definition for \@addtoreset.)

```
93 </2ekernel>
```

\@removefromreset

```
94 <|latexrelease>\IncludeInRelease{2018-04-01}
```

```

95  <{latexrelease}                      {\@removefromreset}{Add interfaces}%
96  <{*2ekernel | latexrelease}%
97  \def\@removefromreset#1#2{%

```

Even though 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.

```

98  \@ifundefined{c@#2}\relax
99  {\begingroup
100   \expandafter\let\csname c@#1\endcsname\@removefromreset
101   \def\@elt##1{%
102     \expandafter\ifx\csname c##1\endcsname\@removefromreset
103     \else
104       \noexpand\@elt{##1}%
105     \fi}%
106   \expandafter\xdef\csname cl@#2\endcsname
107   {\csname cl@#2\endcsname}%
108 \endgroup}%

```

(End of definition for \@removefromreset.)

\@ifbothcounters Test if arg #1 and #2 are counters and if so execute #3.

```

109 \def\@ifbothcounters#1#2#3{%
110   \@ifundefined{c@#1}{\@nocounterr{#1}}%
111   {%
112     \else counter is defined
113     \@ifundefined{c@#2}{\@nocounterr{#2}}%
114     {%
115       \else both counter and within are defined
116       #3}}}

```

(End of definition for \@ifbothcounters.)

```

115 </2ekernel | latexrelease>
116 <{latexrelease}\EndIncludeInRelease>
117 <{latexrelease}\IncludeInRelease{0000-00-00}>
118 <{latexrelease}                      {\@removefromreset}{Add interfaces}%
119 <{latexrelease}\let\@removefromreset\undefined>
120 <{latexrelease}\let\@ifbothcounters\undefined>
121 <{latexrelease}\EndIncludeInRelease>
122 <{*2ekernel}>

```

\counterwithout \counterwithin New implementation using xparse and supporting an optional format argument.

```

123 </2ekernel>
124 <{*2ekernel | latexrelease}>
125 <{latexrelease}\IncludeInRelease{2021/11/15}>
126 <{latexrelease}                  {\counterwithout}{counter without/within}%
127 \NewDocumentCommand \counterwithout {sO{\arabic}mm}{%
128   \@ifbothcounters{#3}{#4}{%
129     \@removefromreset{#3}{#4}%
130     \IfBooleanF {#1}{%
131       \expandafter
132       \gdef\csname the#3\endcsname {#2{#3}}{}}%
133   }%
134 }

```

```

135 \NewDocumentCommand \counterwithin {s0{\arabic}mm}{%
136   \@ifbothcounters{#3}{#4}{%
137     \@addtoreset{#3}{#4}%
138     \IfBooleanF #1{%
139       {\expandafter
140         \gdef\csname the#3\expandafter\endcsname
141         \expandafter
142           {\csname the#4\endcsname .#2{#3}}}}%
143   }%
144 }

145 </2ekernel | latexrelease>
146 <latexrelease>\EndIncludeInRelease
147 <latexrelease>\IncludeInRelease{2018-04-01}
148 <latexrelease>          {\counterwithout}{counter without/within}%
149 <latexrelease>
150 <latexrelease>\def\counterwithout {\@ifstar\counterwithout@s\counterwithout@x}
151 <latexrelease>\def\counterwithout@s#1#2{%
152   <latexrelease>  \@ifbothcounters{#1}{#2}{\@removefromreset{#1}{#2}}}
153 <latexrelease>\def\counterwithout@x#1#2{%
154   <latexrelease>  \@ifbothcounters{#1}{#2}{%
155     <latexrelease>    {\@removefromreset{#1}{#2}}%
156     <latexrelease>    \expandafter
157     <latexrelease>    \gdef\csname the#1\expandafter\endcsname\expandafter
158     <latexrelease>      {\expandafter
159     <latexrelease>        \arabic\csname c@#1\endcsname}}}
160 <latexrelease>
161 <latexrelease>\def\counterwithin{\@ifstar\counterwithin@s\counterwithin@x}
162 <latexrelease>\def\counterwithin@s#1#2{%
163   <latexrelease>  \@ifbothcounters{#1}{#2}{\@addtoreset{#1}{#2}}}

164 <latexrelease>\def\counterwithin@x#1#2{%
165   <latexrelease>  \@ifbothcounters{#1}{#2}{%
166     {\@addtoreset{#1}{#2}}%
167     <latexrelease>    \expandafter
168     <latexrelease>    \gdef\csname the#1\expandafter\endcsname\expandafter
169     <latexrelease>      {\csname the#2\expandafter\endcsname\expandafter
170     <latexrelease>        .\expandafter
171     <latexrelease>        \arabic\csname c@#1\endcsname}}}
172 <latexrelease>
173 <latexrelease>\EndIncludeInRelease
174 <latexrelease>\IncludeInRelease{0000-00-00}
175 <latexrelease>          {\counterwithout}{counter without/within}%
176 <latexrelease>\let \counterwithout \undefined
177 <latexrelease>\let \counterwithout@s \undefined
178 <latexrelease>\let \counterwithout@x \undefined
179 <latexrelease>\let \counterwithin \undefined
180 <latexrelease>\let \counterwithin@s \undefined
181 <latexrelease>\let \counterwithin@x \undefined
182 <latexrelease>\EndIncludeInRelease
183 <*2ekernel>

```

(End of definition for `\counterwithout` and `\counterwithin`.)

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.

```
184 \def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}
```

(End of definition for `\arabic`.)

\roman Representation of `<counter>` as lower-case Roman numerals.

```
185 \def\roman#1{\expandafter\@roman\csname c@#1\endcsname}
```

(End of definition for `\roman`.)

\Roman Representation of `<counter>` as upper-case Roman numerals.

```
186 \def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}
```

(End of definition for `\Roman`.)

\alph Representation of `<counter>` as a lower-case letter: 1 = a, 2 = b, etc.

```
187 \def\alph#1{\expandafter\@alph\csname c@#1\endcsname}
```

(End of definition for `\alph`.)

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

```
188 \def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}
```

(End of definition for `\Alph`.)

\fnsymbol Representation of `<COUNTER>` as a footnote symbol: 1 = *, 2 = †, etc.

```
189 \def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}
```

(End of definition for `\fnsymbol`.)

\@arabic \@arabic\FOOcounter Representation of `\FOOcounter` as arabic numerals.

```
190 \def\@arabic#1{\number #1} %% changed 29 Apr 86
```

(End of definition for `\@arabic`.)

\@roman \@roman\FOOcounter Representation of `\FOOcounter` as lower-case Roman numerals.

```
191 \def\@roman#1{\romannumeral #1}
```

(End of definition for `\@roman`.)

\@Roman \@Roman\FOOcounter Representation of `\FOOcounter` as upper-case Roman numerals.

```
192 \def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}
```

(End of definition for `\@Roman`.)

\@slowromancap Fully expandable macro to change a roman number to uppercase.

```
193 \def\@slowromancap#1{\ifx @#1% then terminate
194   \else
195     \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
196       c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi\fi
197     \expandafter\@slowromancap
198   \fi
199 }
```

(End of definition for `\@slowromancap`.)

\@alph \c@alph\FOOcounter Representation of \FOOcounter as a lower-case letter: 1 = a, 2 = b, etc.

```
200 \def\@alph#1{%
201   \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
202   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
203   y\or z\else\@ctrerr\fi}
```

(End of definition for \@alph.)

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

```
204 \def\@Alph#1{%
205   \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
206   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
207   Y\or Z\else\@ctrerr\fi}
```

(End of definition for \@Alph.)

\@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 L^AT_EX (as recommended) this unfortunate side effect is not present.

```
208 </2ekernel>
209 <|latexrelease|\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TextOrMath}%
210 <*2ekernel | latexrelease>
211 \def\@fnsymbol#1{%
212   \ifcase#1\or \TextOrMath{textasteriskcentered}*\or
213   \TextOrMath{textdagger} \dagger\or
214   \TextOrMath{textdaggerdbl} \ddagger\or
215   \TextOrMath{textsection} \mathsection\or
216   \TextOrMath{textparagraph} \mathparagraph\or
217   \TextOrMath{textbardbl} \|\or
218   \TextOrMath{\textasteriskcentered}\textasteriskcentered\{**}\or
219   \TextOrMath{\textdagger}\textdagger\{ \dagger\dagger\}\or
220   \TextOrMath{\textdaggerdbl}\textdaggerdbl\{ \ddagger\ddagger\}\else
221   \@ctrerr\fi
222 }%
223 </2ekernel | latexrelease>
224 <|latexrelease|\EndIncludeInRelease
225 <|latexrelease|\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TextOrMath}%
226 <|latexrelease|\def\@fnsymbol#1{\ensuremath{%
227 <|latexrelease> \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
228 <|latexrelease> \mathparagraph\or \|\or **\or \dagger\dagger}
```

```

229 <{latexrelease}>      \or \ddagger\ddagger \else\ctrerr\fi}}}%
230 <{latexrelease}>\EndIncludeInRelease
231 <{*2ekernel}>

```

(End of definition for `\@fnsymbol`.)

- \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 will have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of eTeX will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running eTeX but making sure not to permanently turn `\protected` into `\relax`.

```

232 </2ekernel>
233 <{latexrelease}>\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
234 <{*2ekernel | latexrelease}>
235 \begingroup\expandafter\expandafter\expandafter\endgroup
236 \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.

```

237 \DeclareRobustCommand\TextOrMath{%
238   \ifmmode \expandafter\@secondoftwo
239   \else \expandafter\@firstoftwo \fi}
240 \protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
241 \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.

```

242 \protected\expandafter\def\csname TextOrMath\space\endcsname{%
243   \ifmmode \expandafter\@secondoftwo
244   \else \expandafter\@firstoftwo \fi}
245 \edef\TextOrMath#1#2{%
246   \expandafter\noexpand\csname TextOrMath\space\endcsname
247   {#1}{#2}}
248 \fi
249 </2ekernel | latexrelease>
250 <{latexrelease}>\EndIncludeInRelease
251 <{latexrelease}>\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
252 <{latexrelease}>\let\TextOrMath\@undefined
253 <{latexrelease}>\EndIncludeInRelease
254 <{*2ekernel}>

```

(End of definition for `\TextOrMath`.)

```

255 </2ekernel>

```

File 23

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.  
 1  <*2ekernel>  
 2  \message{lengths,}  
  
\newlength  
 3  \def\newlength#1{\@ifdefinable#1{\newskip#1}}  
  
(End of definition for \newlength.)  
  
\setlength  
 4  </2ekernel>  
 5  <latexrelease>\IncludeInRelease{2015/01/01}%  
 6  <latexrelease>          {\setlength}{Using \setlength with \dimen0}%  
 7  <*2ekernel | latexrelease>  
 8  \def\setlength#1#2{#1 #2\relax}  
 9  </2ekernel | latexrelease>  
10 <latexrelease>\EndIncludeInRelease  
11 <latexrelease>\IncludeInRelease{0000/00/00}%  
12 <latexrelease>          {\setlength}{Using \setlength with \dimen0}%  
13 <latexrelease>\def\setlength#1#2{#1#2\relax}  
14 <latexrelease>\EndIncludeInRelease  
15 <*2ekernel>  
  
(End of definition for \setlength.)  
  
\addtolength \relax added 24 Mar 86  
 16 \def\addtolength#1#2{\advance#1 #2\relax}  
  
(End of definition for \addtolength.)  
  
\settoheight The obvious analogs of \settowidth.  
 \settodepth  
 \settowidth  
 \settodim  
 17 </2ekernel>  
 18 <*2ekernel | latexrelease>  
 19 <latexrelease>\IncludeInRelease{2024/11/01}%  
 20 <latexrelease>          {\@settodim}{suspend tagging}%  
 21 \def\@settodim#1#2#3{\setbox\@tempboxa\hbox  
 22 {{\SuspendTagging{\@settodim}#3\ResumeTagging{\@settodim}}}}#2#1\@tempboxa  
 Clear the memory afterwards (which might be a lot).  
 23   \setbox\@tempboxa\box\voidb@x  
 24 </2ekernel | latexrelease>  
 25 <latexrelease>\EndIncludeInRelease
```

```

26 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
27 〈\latexrelease〉                      {\@settodim}{suspend tagging}%
28 〈\latexrelease〉\def\@settodim#1#2#3{\setbox\@tempboxa\hbox{\#3}#2#1\@tempboxa
29 〈\latexrelease〉                  \setbox\@tempboxa\box\voidb@x}
30 〈\latexrelease〉\EndIncludeInRelease
31 〈*2ekernel〉

32 \DeclareRobustCommand\settoheight{\@settodim\ht}
33 \DeclareRobustCommand\settodepth {\@settodim\dp}
34 \DeclareRobustCommand\settowidth {\@settodim\wd}

(End of definition for \settoheight and others.)

```

\@settopoint This macro takes the contents of the skip register that is supplied as its argument and removes the fractional part to make it a whole number of points. This can be used in class files to avoid values like 345.4666666pt when calculating a dimension.

```

35 \def\@settopoint#1{\divide#1\p@\multiply#1\p@}
36 〈/2ekernel〉

```

(*End of definition for \@settopoint.*)

File 24

ltfssbas.dtx

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

Warning: The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

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.

```
1  {*2ekernel}
2  \def\@nomath#1{\relax\ifmmode
3      \font@warning{Command \noexpand#1invalid in math mode}\fi}
```

(End of definition for `\@nomath`.)

`\no@alphabet@error` The macro `\no@alphabet@error` is called whenever the user requests a math *alphabet* that is not available in the current *version*. In math mode an error message is produced otherwise the command keeps silent. The argument is the name of the control sequence that identifies the math *alphabet*. The `\relax` at the beginning is necessary to prevent T_EX from scanning too far in certain situations.

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

(End of definition for `\no@alphabet@error`.)

`\new@mathgroup` `\mathgroup` We also give a new name to `\newfam` and `\fam` to avoid verbal confusion (see the introduction).³¹

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

(End of definition for `\new@mathgroup` and `\mathgroup`.)

³¹For 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:  
18  \def\DeclareFontShape{\begingroup  
First we restore the catcodes of all characters used in the syntax.  
19    \nfss@catcodes  
We use \expandafter \endgroup to restore catcode in case something goes wrong with  
the argument parsing (suggested by Tim Van Zandt)  
20    \expandafter\endgroup  
21    \DeclareFontShape@}  
(End of definition for \DeclareFontShape.)
```

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

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

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

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

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

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

```

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

```

(End of definition for `\DeclareFontShape@`.)

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

```

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

```

(End of definition for `\DeclareFixedFont`.)

`\do@subst@correction`

```

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

```

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

```

91  \aftergroup\subst@correction
92  }

```

(End of definition for `\do@subst@correction`.)

```
\DeclareFontFamily
```

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

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

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

Now we have to define the macro `\(#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`.

```
105 \def\reserved@a{#3}%
```

We compare `\reserved@a` with `\@empty`. If these two are the same we `\let` the ‘extra’ macro equal to `\@empty` which is not the same as doing a `\let` to `\reserved@a` — the latter would blow one extra memory location rather than reusing the one from `\@empty`.

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

(End of definition for `\DeclareFontFamily`.)

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

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

(End of definition for `\cdp@list`.)

```
\cdp@elt
```

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

(End of definition for `\cdp@elt`.)

```
\DeclareFontEncoding
```

```
118 \def\DeclareFontEncoding{%
```

First we start with ignoring all blanks and newlines since every surplus space in the second or third argument will come out in a weird place in the document.

```

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

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

```

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

```

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

```

Keep a record of the last encoding being declared:

```

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

```

(End of definition for `\DeclareFontEncoding`.)

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

```
140 \def\LastDeclaredEncoding{}%
```

(End of definition for `\LastDeclaredEncoding`.)

`\DeclareFontSubstitution`

```

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

```

We loop through the `\cdp@list` and rebuild it anew in `\toks@` thereby replacing the defaults for the encoding in question with the new defaults. It is important to store the encoding to test against expanded in `\reserved@a` since it might just be `\LastDeclaredEncoding` that is passed as #1.

```

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

```

Here we use the new defaults but we use ##1 (i.e., the encoding name already stored previously) since we know that it is expanded.

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

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

```
154     \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
155     \fi}%
156     \cdp@list
157     \xdef\cdp@list{\the\toks@}%
158     \endgroup
159     \global
160     \cnamedef{D@#1}{%
161         \def\default@family{#2}%
162         \def\default@series{#3}%
163         \def\default@shape{#4}%
164     }%
165     \fi
166 }
167 \onlypreamble\DeclareFontSubstitution
```

(End of definition for \DeclareFontSubstitution.)

\DeclareFontEncodingDefaults

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

(End of definition for \DeclareFontEncodingDefaults.)

\default@T

\default@M

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

(End of definition for \default@T and \default@M.)

\DeclarePreloadSizes

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

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

```
189 \begingroup
```

We define a macro `\reserved@f`³² that grabs the next *size* and loads the corresponding font. This is done by delimiting `\reserved@f`'s only argument by the token `,` (comma).

```
190 \def\reserved@f##1,{%
```

The end of the list will be detected when there are no more elements, i.e. when `\reserved@f`'s argument is empty. The trick used here is explained in Appendix D of the TeXbook: if the argument is empty the `\if` will select the first clause and `\let` `\reserved@f` equal to `\relax`. (We use the `>` character here since it cannot appear in font file names.)

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

Otherwise, we define `\font@name` appropriately and call `\pickup@font` to do the work. Note that the requested `\curr@fontshape` combination must have been defined, or you will get an error. The definition of `\font@name` is carried out globally to be consistent with the rest of the code in this file.

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

Now we forget the name of the font just loaded. More precisely, we set the corresponding control sequence to `\relax`. This means that later on, when the font is first used, the macro `\define@newfont` is called again to execute the 'extra' macro for this font.

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

Finally we call `\reserved@f` again to process the next *size*. If `\reserved@f` was `\let` equal to `\relax` this will end the macro.

```
198 \reserved@f}%
```

We finish with reinserting the list of sizes after the `\reserved@f` macro and appending an empty element so that the end of the list is recognized properly.

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

(End of definition for `\DeclarePreloadSizes`.)

`\ifmath@fonts` We need a switch to decide if we have to switch math fonts. For this purpose we provide `\ifmath@fonts` that can be set to true or false by the `\S@...` macros depending on if math fonts are provided for this size or not. The default is of course to switch all fonts.

```
204 \newif\ifmath@fonts \math@fontstrue
```

(End of definition for `\ifmath@fonts`.)

³²We cannot use `\@tempa` since it is needed in `\pickup@font`.

\DeclareMathSizes \DeclareMathSizes takes the text size, math text size, math script size, and math scriptscript size as arguments and defines the right \S@... macro.

```

205 \def\DeclareMathSizes{%
206   \@ifstar{\@DeclareMathSizes\math@fontsfalse}{%
207     {\@DeclareMathSizes{}}}
208   \onlypreamble\DeclareMathSizes

```

(End of definition for \DeclareMathSizes and \DeclareMathSizes*.)

\@DeclareMathSizes This modification by Michael J. Downes on comp.text.tex on 2002/10/17 allows the user to have settings such as

```

\DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}.

```

```

209 </2ekernel>
210 <latexrelease>\IncludeInRelease[2015/01/01]{\@DeclareMathSizes}%
211 <latexrelease>                                {Arbitrary units in \DeclareMathSizes}%
212 <*2ekernel | latexrelease>
213 \def\@DeclareMathSizes #1#2#3#4#5{%
214   \@defaultunits\dimen@ #2pt\relax\@nnil
215   \if $#3$%
216     \expandafter\let\csname S@\strip@pt\dimen@\endcsname\math@fontsfalse
217   \else
218     \@defaultunits\dimen@ii #3pt\relax\@nnil
219     \@defaultunits\@tempdima #4pt\relax\@nnil
220     \@defaultunits\@tempdimb #5pt\relax\@nnil
221     \toks@{\#1}%
222     \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
223       \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
224       \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
225       \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
226       \the\toks@
227     }%
228   \fi
229 }%
230 </2ekernel | latexrelease>
231 <latexrelease>\EndIncludeInRelease
232 <latexrelease>\IncludeInRelease[0000/00/00]{\@DeclareMathSizes}%
233 <latexrelease>                                {Arbitrary units in \DeclareMathSizes}%
234 <latexrelease>\def\@DeclareMathSizes#1#2#3#4#5{%
235   \@defaultunits\dimen@#2pt\relax\@nnil
236   \if $#3$%
237     \expandafter \let
238     \csname S@\strip@pt\dimen@\endcsname
239     \math@fontsfalse
240   \else
241     \expandafter \gdef
242     \csname S@\strip@pt\dimen@\endcsname
243     {\gdef\tf@size{\#3}\gdef\sf@size{\#4}%
244      \gdef\ssf@size{\#5}%
245     \#1%
246   \fi}%
247   \expandafter \gdef
248   \csname S@\strip@pt\dimen@\endcsname
249   {\gdef\tf@size{\#3}\gdef\sf@size{\#4}%

```

```
250 \onlypreamble\@DeclarMathSizes
```

(End of definition for \@DeclarMathSizes.)

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.

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

If the new encoding is the same as the old encoding we have nothing to do. However, in case we had a sequence of several encoding changes without a `\selectfont` in-between we can save processing by making sure that `\enc@update` is `\relax`.

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

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

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

(End of definition for `\fontencoding` and `\f@encoding`.)

```
\@enc@update
```

```
263 \def\@enc@update{%
```

When `\@enc@update` is executed `\f@encoding` holds the encoding name for the new encoding and `\cf@encoding` the name of the last active encoding.

We start by setting the init command for encoding dependent macros to `\@changed@cmd`.

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

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

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

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

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

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

```

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

(End of definition for \@@enc@update.)

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

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

(End of definition for \enc@update.)

```
\fontfamily
\f@family
\fontseries
\f@series
\fontshape
\f@shape
```

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

There are now defined later (and differently).

```
280 \%Declarerobustcommand\fontseries[1]{\edef\f@series{#1}}
281 \%Declarerobustcommand\fontshape [1]{\edef\f@shape{#1}}
```

(End of definition for \fontfamily and others.)

`\usefont` Some handy abbreviation if you want to get some particular font in the current size. If also the size should change one has to issue a `\fontsize` command first.

`\fontencoding` needs to do some setup work so we call that, but instead of calling `\fontfamily`, `\fontseries` and `\fontshape` it earlier versions of this code did, we now set `\f@family`, etc. directly. If we would call `\fontseries` or `\fontshape` as it was done in the past, they would now interact with the existing series and shape which is not desired if we intend to use an explicit font shape!

```

282 </2ekernel>
283 <*2ekernel | latexrelease>
284 <latexrelease>\IncludeInRelease{2021/06/01}%
285 <latexrelease>          {\usefont}{Force font face}%
286 \Declarerobustcommand\usefont[4]{\fontencoding{#1}%
287   \edef\f@family{#2}%
288   \set@target@series{#3}%
289   \edef\f@shape{#4}}%
```

Any earlier `\fontseries`, etc. should be canceled and we should switch unconditionally to the requested font face so we drop any code that may have been stored in `\delayed@f@adjustment`.

```

290 \let\delayed@f@adjustment\empty
291 \selectfont
292 \ignorespaces}
293 </2ekernel | latexrelease>
294 <latexrelease>\EndIncludeInRelease
295 <latexrelease>\IncludeInRelease{2020/02/02}%
296 <latexrelease>          {\usefont}{Drop m in usefont}%
297 <latexrelease>
298 <latexrelease>\Declarerobustcommand\usefont[4]{\fontencoding{#1}%
299 <latexrelease>  \edef\f@family{#2}%
300 <latexrelease>  \set@target@series{#3}%
301 <latexrelease>  \edef\f@shape{#4}\selectfont
```

```

302 <|latexrelease>    \ignorespaces}
303 <|latexrelease>
304 <|latexrelease>\EndIncludeInRelease
305 <|latexrelease>\IncludeInRelease{0000/00/00}%
306 <|latexrelease>                {\usefont}{Drop m in usefont}%
307 <|latexrelease>
308 <|latexrelease>\DeclareRobustCommand\usefont[4]{\fontencoding{\#1}%
309 <|latexrelease>    \edef\f@family{\#2}%
310 <|latexrelease>    \edef\f@series{\#3}%
311 <|latexrelease>    \edef\f@shape{\#4}\selectfont
312 <|latexrelease>    \ignorespaces}
313 <|latexrelease>
314 <|latexrelease>\EndIncludeInRelease
315 {*2ekernel}

```

(End of definition for `\usefont`.)

\linespread The command `\linespread` changes the current `\baselinestretch` by calling `\set@fontsize`. The values for `\f@size` and `\f@baselineskip` will be left unchanged.

```

316 \DeclareRobustCommand\linespread[1]
317   {\set@fontsize{\#1}\f@size\f@baselineskip}

```

(End of definition for `\linespread`.)

\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.

```

318 \DeclareRobustCommand\fontsize[2]
319   {\set@fontsize\baselinestretch{\#1}{\#2}}

```

(End of definition for `\fontsize`.)

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

```

320 \let\f@family\empty
321 \let\f@series\empty
322 \let\f@shape\empty
323 \let\f@size\empty
324 \let\f@baselineskip\empty
325 \let\f@linespread\empty

```

(End of definition for `\f@linespread`.)

\cf@encoding

```

326 \let\f@encoding\empty
327 \let\cf@encoding\empty

```

(End of definition for `\cf@encoding`.)

- \@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.
- ```
328 \def\@defaultunits{\afterassignment\remove@to@nnil}
```
- (End of definition for \@defaultunits.)
- \strip@pt This macro strips the characters pt produced by using \the on a dimen register.
- ```
329 \begingroup
330 \catcode`P=12
331 \catcode`T=12
332 \lowercase{
333 \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
334 \expandafter\endgroup\x
335 \def\strip@pt{\expandafter\rem@pt\the}
```
- (End of definition for \strip@pt and \rem@pt.)
- \mathversion \mathversion takes the math *version* name as argument, defines \math@version appropriately and switches to the font selected forcing a call to \glb@settings if the *version* is known to the system.
- ```
336 \DeclareRobustCommand\mathversion[1]
337   {\@nomath\mathversion
338   \expandafter\ifx\csname mv@#1\endcsname\relax
339     \@latex@error{Math version '#1' is not defined}\@eha\else
```
- If there has been a frozen math version reset locally. See GH 1028.
- ```
340 \ifcsname mv@\math@version @frozen\endcsname
341 \expandafter\let
342 \csname mv@\math@version @frozen\expandafter\endcsname
343 \csname mv@\math@version\endcsname
344 \fi
345 \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.
- ```
346   \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.
- ```
347 \aftergroup\glb@settings
348 \fi}
```

(End of definition for `\mathversion` and `\math@version`.)

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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 2<sub>C</sub> we need to implement it in a somewhat slower way.

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

```
\frozen@everymath New internal names for \everymath and \everydisplay.
\frozen@everydisplay
349 \let\frozen@everymath\everymath
350 \let\frozen@everydisplay\everydisplay
```

(End of definition for `\frozen@everymath` and `\frozen@everydisplay`.)

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

(End of definition for `\everymath` and `\everydisplay`.)

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

The check code may push tokens after the math formula with `\aftergroup` and they would prevent a \$\$ from dropping following spaces. We therefore use a switch to be set as the first thing after the group so that following code can determine if there was a display or some inline math (in the latter case we better not drop spaces). After setting the switch we also have to place `\ignorespaces` because setting the switch may be the only thing that happens after the display. The issue with handling of spaces was found in 2022, but it is really a bug fix for the code added in 2021/11.

```
353 </2ekernel>
354 <latexrelease>\IncludeInRelease{2021/11/15}
355 <latexrelease> {\frozen@everydisplay}{Handle spaces after math} %
356 <*2ekernel | latexrelease>
357 \frozen@everydisplay = {
358 \aftergroup\@ignoretrue \aftergroup\ignorespaces
359 \check@mathfonts
360 \the\everydisplay}
```

(End of definition for `\frozen@everydisplay`.)

```
\frozen@everymath The frozen code for inline math is similar, except that here we do not want to drop following spaces.
```

```
361 \frozen@everymath = {
362 \aftergroup\@ignorefalse
363 \check@mathfonts
364 \the\everymath}
```

```

(End of definition for \frozen@everymath.)
```

```

365 </2ekernel | latexrelease>
366 <latexrelease>\EndIncludeInRelease
367 <latexrelease>\IncludeInRelease{2020/10/01}
368 <latexrelease> {\frozen@everydisplay}{Handle spaces after math}%
369 <latexrelease>
370 <latexrelease>\frozen@everydisplay = {\check@mathfonts
371 <latexrelease> \the\everydisplay}
372 <latexrelease>\frozen@everymath = {\check@mathfonts
373 <latexrelease> \the\everymath}
374 <latexrelease>
375 <latexrelease>\EndIncludeInRelease
376 <*2ekernel>
```

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

```

377 \let\curr@math@size\empty
```

```
(End of definition for \curr@math@size.)
```

### 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.

```

378 \def\pickup@font{%
379 \expandafter \ifx \font@name \relax
380 \define@newfont
381 \fi}
```

```
(End of definition for \pickup@font.)
```

\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

```

382 {\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.

```

383 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}%
384 \def\f@family{#2}%
385 \def\f@series{#3}%
386 \def\f@shape{#4}%
387 \def\f@size{#5}}}
```

```
(End of definition for \split@name.)
```

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

```

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

```
(End of definition for \curr@fontshape.)
```

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

```
389 \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.

```
390 \begingroup
391 \let\typeout\@font@info
392 \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`.

```
393 \expandafter\expandafter\expandafter
394 \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.

```
395 % \expandafter\ifx
396 % \csname\curr@fontshape\endcsname \relax
397 \try@load@fontshape % try always
398 % \fi
399 \expandafter\ifx
400 \csname\curr@fontshape\endcsname \relax
401 \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).

```
402 % \csname\curr@fontshape\endcsname
403 \extract@font\fi
```

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

```
404 \endgroup}
405 \def\try@load@fontshape{%
406 \expandafter
407 \ifx\csname \f@encoding+\f@family\endcsname\relax
408 \font@info{Trying to load font information for
409 \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.

```
410 \global\expandafter\let
411 \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).

```
412 \nfss@catcodes
413 \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.

```

414 \edef\reserved@a{%
415 \lowercase{%
416 \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}{}}%
417 \reserved@a\relax
418 {\@input{\f@encoding\f@family.fd}}{}}%
419 \fi}

```

(End of definition for `\define@newfont`.)

- `\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.

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

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

```

421 \makeatletter
422 \catcode`\@=10
423 \catcode`^I=10
424 \catcode`^M=10

```

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

```

425 \catcode`\\=0
426 \catcode`{\=10
427 \catcode`}=10
428 \catcode`#=6
429 \catcode`^=7
430 \catcode`^@=14

```

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

```

431 \@makeother<%
432 \@makeother>%
433 \@makeother*%
434 \@makeother.%%
435 \@makeother-%
436 \@makeother/%
437 \@makeother[%
438 \@makeother]%
439 \@makeother`%
440 \@makeother'%
441 \@makeother"%
442 }

```

(End of definition for `\nfss@catcodes`.)

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

```
443 </2ekernel>
444 <*2ekernel | latexrelease>
445 <latexrelease>\IncludeInRelease{2020/02/02}%
446 <latexrelease> {\LoadFontDefinitionFile}{Loading .fd files}%
447 \def\LoadFontDefinitionFile#1#2{%
448 \begingroup
449 \edef\f@encoding{#1}%
450 \edef\f@family{#2}%
451 \try@load@fontshape
452 \endgroup
453 }
454 </2ekernel | latexrelease>
455 <latexrelease>\EndIncludeInRelease
456 <latexrelease>\IncludeInRelease{0000/00/00}%
457 <latexrelease> {\LoadFontDefinitionFile}{Loading .fd files}%
458 <latexrelease>
459 <latexrelease>\let\LoadFontDefinitionFile@\undefined
460 <latexrelease>\EndIncludeInRelease
461 <*2ekernel>
```

(End of definition for `\LoadFontDefinitionFile`.)

`\DeclareFontFamilySubstitution` The idea for this macro is stolen from the `substitutefont` package by Günter Milde, with some modifications and a new name.

Its purpose is to provide characters in a special encoding that are not available in the current font family to be taken from a different family that is visually compatible (or not if you choose badly). For example, you can match the GFS Didot Greek characters with TeX Gyre Pagella (Palatino) by specifying

```
\DeclareFontFamilySubstitution{LGR}{qpl}{udidot}
```

This way if you ask for the LGR encoding in for the qpl family you get the characters from the uidot 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).

```
462 </2ekernel>
463 <*2ekernel | latexrelease>
464 <latexrelease>\IncludeInRelease{2020/02/02}%
465 <latexrelease> {\DeclareFontFamilySubstitution}{Provide family substitution}%
466 \begingroup
467 \nfss@catcodes
468 \gdef\DeclareFontFamilySubstitution#1#2#3{%
```

We only provide a set of silent substitutions. The package also (re)declared the family, but this is incorrect in my eyes and it is better to handle that differently.

Of course the families may still need loading at this point and so we arrange for this. Otherwise we might run into trouble because the necessary `\DeclareFontFamily` has not been seen.

```
469 \LoadFontDefinitionFile{#1}{#2}%
470 \LoadFontDefinitionFile{#1}{#3}%
```

```

471 \DeclareFontShape{\#1}{\#2}{m}{it}{<->ssub * #3/m/it}{}%
472 \DeclareFontShape{\#1}{\#2}{m}{n}{<->ssub * #3/m/n}{}%
473 \DeclareFontShape{\#1}{\#2}{m}{sc}{<->ssub * #3/m/sc}{}%
474 \DeclareFontShape{\#1}{\#2}{m}{s1}{<->ssub * #3/m/s1}{}%

```

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.

```

475 \DeclareFontShape{\#1}{\#2}{m}{sw}{<->ssub * #3/m/sw}{}%
476 \DeclareFontShape{\#1}{\#2}{m}{scit}{<->ssub * #3/m/scit}{}%
477 \DeclareFontShape{\#1}{\#2}{m}{scsl}{<->ssub * #3/m/scsl}{}%

```

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

```

478 \DeclareFontShape{\#1}{\#2}{b}{it}{<->ssub * #3/b/it}{}%
479 \DeclareFontShape{\#1}{\#2}{b}{n}{<->ssub * #3/b/n}{}%
480 \DeclareFontShape{\#1}{\#2}{b}{scit}{<->ssub * #3/b/scit}{}%
481 \DeclareFontShape{\#1}{\#2}{b}{scsl}{<->ssub * #3/b/scsl}{}%
482 \DeclareFontShape{\#1}{\#2}{b}{sc}{<->ssub * #3/b/sc}{}%
483 \DeclareFontShape{\#1}{\#2}{b}{s1}{<->ssub * #3/b/s1}{}%
484 \DeclareFontShape{\#1}{\#2}{b}{sw}{<->ssub * #3/b/sw}{}%
485 \DeclareFontShape{\#1}{\#2}{bx}{it}{<->ssub * #3/bx/it}{}%
486 \DeclareFontShape{\#1}{\#2}{bx}{n}{<->ssub * #3/bx/n}{}%
487 \DeclareFontShape{\#1}{\#2}{bx}{scit}{<->ssub * #3/bx/scit}{}%
488 \DeclareFontShape{\#1}{\#2}{bx}{scsl}{<->ssub * #3/bx/scsl}{}%
489 \DeclareFontShape{\#1}{\#2}{bx}{sc}{<->ssub * #3/bx/sc}{}%
490 \DeclareFontShape{\#1}{\#2}{bx}{s1}{<->ssub * #3/bx/s1}{}%
491 \DeclareFontShape{\#1}{\#2}{bx}{sw}{<->ssub * #3/bx/sw}{}%
492 }
493 \endgroup
494 </2ekernel | latexrelease>
495 <latexrelease>\EndIncludeInRelease
496 <latexrelease>\IncludeInRelease{0000/00/00}%
497 <latexrelease> {\DeclareFontFamilySubstitution}{Provide family substitution}%
498 <latexrelease>
499 <latexrelease>\let\DeclareFontFamilySubstitution@\undefined
500 <latexrelease>\EndIncludeInRelease
501 <*2ekernel>

```

(End of definition for \DeclareFontFamilySubstitution.)

**\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!

```

502 </2ekernel>
503 <*2ekernel | latexrelease>
504 <latexrelease>\IncludeInRelease{2019/10/01}%
505 <latexrelease> {\DeclareErrorFont}{No side effects please}%
506 \def\DeclareErrorFont#1#2#3#4#5{%
507 \xdef\error@fontshape{%
508 \noexpand\expandafter\noexpand\split@name\noexpand\string
509 \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
510 \noexpand\@nil}%

```

Initialize all those internal variables which may or may not have values in the first seconds of NFSS' bootstrapping process. Later on such values will be updated when an encoding is selected, etc.

We definitely don't want to set `\f@encoding`; we can set all the others since if they are left "blank" any selection would grab "error default values" as well. However, this probably should go also—and now it did.

```

511 % \gdef\f@encoding{#1}%
512 \gdef\default@family{#2}%
513 \gdef\default@series{#3}%
514 \gdef\default@shape{#4}%
515 }
516 </2ekernel | latexrelease>
517 <latexrelease>\EndIncludeInRelease
518 <latexrelease>\IncludeInRelease{0000/00/00}%
519 <latexrelease> {\DeclareErrorFont}{No side effects please}%
520 <latexrelease>
521 <latexrelease>\def\DeclareErrorFont#1#2#3#4#5{%
522 <latexrelease> \xdef\error@fontshape{%
523 <latexrelease> \noexpand\expandafter\noexpand\split@name\noexpand\string
524 <latexrelease> \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
525 <latexrelease> \noexpand\@nil}%
526 <latexrelease> \gdef\default@family{#2}%
527 <latexrelease> \gdef\default@series{#3}%
528 <latexrelease> \gdef\default@shape{#4}%
529 <latexrelease> \global\let\f@family\default@family
530 <latexrelease> \global\let\f@series\default@series
531 <latexrelease> \global\let\f@shape\default@shape
532 <latexrelease> \gdef\f@size{#5}%
533 <latexrelease> \gdef\f@baselineskip{#5pt}%
534 <latexrelease>}
535 <latexrelease>\EndIncludeInRelease
536 <*2ekernel>
537 \onlypreamble\DeclareErrorFont

```

(*End of definition for `\DeclareErrorFont`.*)

`\wrong@fontshape` Before we come to the macro `\extract@font` we have to take care of unknown `\curr@fontshape` combinations. The general strategy is to issue a warning and to try a default *shape*, then a default *series*, and finally a default *family*. If this last one also fails T<sub>E</sub>X will go into an infinite loop. But if the defaults are set incorrectly one deserves nothing else!

```

538 </2ekernel>
539 <latexrelease>\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
540 <latexrelease> {Font substitution in preamble}%
541 <*2ekernel | latexrelease>
542 \def\wrong@fontshape{%
543 \csname D@\f@encoding\endcsname % install defaults if in math

```

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

```

544 \edef\reserved@a{\csname\curr@fontshape\endcsname}%
545 \ifx\last@fontshape\reserved@a
546 \errmessage{Corrupted NFSS tables}%
547 \error@fontshape
548 \else

```

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

```

549 \let\f@shape\default@shape

```

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

```
550 \expandafter\ifx\csname\curr@fontshape\endcsname\relax
551 \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!

```
552 \expandafter
553 \ifx\csname\curr@fontshape\endcsname\relax
554 \let\f@family\default@family
```

If we change the font family and we are in the preamble then the corresponding .fd file may not been loaded yet. Therefore we try this now. Otherwise equating the requested font shape with the finally selected fontshape below will fail and can result in “NFSS tables corrupted”. After begin document that will not happen as all .fd files involved in substitution are loaded at \begin{document}.

```
555 \begingroup
556 \try@load@fontshape
557 \endgroup
558 \fi \fi
559 \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.

```
560 \@font@warning{Font shape `\\expandafter\\string\\reserved@a'
561 \\expandafter\\gobble\\string\\undefined\\MessageBreak
562 using `\\curr@fontshape' instead\\@wrong@font@char}%
563 \\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.

```
564 \\gdef\\defaultsubs{%
565 \\@font@warning{Some font shapes were not available, defaults
566 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.

```
567 \\global\\expandafter\\expandafter\\expandafter\\let
568 \\expandafter\\reserved@a
569 \\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.

```

570 \xdef\font@name{%
571 \csname\curr@fontshape/\f@size\endcsname}%
572 \pickup@font}
573 {/2ekernel | latexrelease}
574 \langle latexrelease\rangle\EndIncludeInRelease
575 \langle latexrelease\rangle\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
576 \langle latexrelease\rangle {Font substitution in preamble}%
577 \langle latexrelease\rangle\def\wrong@fontshape{%
578 \langle latexrelease\rangle \csname D@\f@encoding\endcsname
579 \langle latexrelease\rangle \edef\reserved@a{\csname\curr@fontshape\endcsname}%
580 \langle latexrelease\rangle \ifx\last@fontshape\reserved@a
581 \errmessage{Corrupted NFSS tables}%
582 \langle latexrelease\rangle \error@fontshape
583 \langle latexrelease\rangle \else
584 \langle latexrelease\rangle \let\f@shape\default@shape
585 \langle latexrelease\rangle \expandafter\ifx\csname\curr@fontshape\endcsname\relax
586 \langle latexrelease\rangle \let\f@series\default@series
587 \expandafter
588 \langle latexrelease\rangle \expandafter
589 \langle latexrelease\rangle \ifx\csname\curr@fontshape\endcsname\relax
590 \langle latexrelease\rangle \let\f@family\default@family
591 \fi \fi
592 \langle latexrelease\rangle \fi
593 \langle latexrelease\rangle \expandafter\string\reserved@a'
594 \langle latexrelease\rangle \expandafter\@gobble\string\@undefined
595 \langle latexrelease\rangle \MessageBreak
596 \langle latexrelease\rangle using '\curr@fontshape' instead\@wrong@font@char}%
597 \langle latexrelease\rangle \global\let\last@fontshape\reserved@a
598 \langle latexrelease\rangle \gdef\@defaultsubs{%
599 \langle latexrelease\rangle \expandafter\warning{Some font shapes were not available,
600 \langle latexrelease\rangle defaults substituted.\@gobbletwo}}%
601 \langle latexrelease\rangle \global\expandafter\expandafter\expandafter\let
602 \expandafter\reserved@a
603 \langle latexrelease\rangle \csname\curr@fontshape\endcsname
604 \langle latexrelease\rangle \xdef\font@name{%
605 \langle latexrelease\rangle \csname\curr@fontshape/\f@size\endcsname}%
606 \langle latexrelease\rangle \pickup@font}
607 \langle latexrelease\rangle\EndIncludeInRelease
608 {*2ekernel}

```

(End of definition for `\wrong@fontshape`.)

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

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

(End of definition for `\@wrong@font@char`.)

`\@@defaultsubs` See above.

```
610 \let\@defaultsubs\relax
```

(End of definition for `\@@defaultsubs` and `\@defaultsubs`.)

`\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).

611 `\def\strip@prefix#1>{}`

(End of definition for `\strip@prefix`.)

## 4 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

612 `\let\install@mathalphabet\gdef`

(End of definition for `\install@mathalphabet`.)

`\math@fonts`

613 `\let\math@fonts\empty`

(End of definition for `\math@fonts`.)

`\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.

614 `%\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.

615 `% \begingroup`

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

616 `% \escapechar\m@ne`

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

We globally select the math fonts...

618 `% \globaldefs\one \math@fonts`

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

619 `% \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.

620 `% \xdef#1{\noexpand\use@mathgroup\noexpand#2%`

621 `% {\number\csname c@mv@\math@version\endcsname}}%`

But this is not sufficient, as we learned the hard way. The problem here is that the loading of the fonts that comprise the alphabet identifier #1, as well as the necessary math font assignments is deferred until it is used. This is OK so far, but if the fonts are switched within the current formula (which may happen if a sub-formula is a box that contains a math version switch) the font assignments for #1 are not restored unless #1 is used again. This is disastrous since TeX sees the wrong fonts at the end of the math formula, when it converts the math list into a horizontal list.

This is taken into account as follows: When a math alphabet identifier is used for the first time in a certain version it modifies the corresponding macro `\mv@<version>` so that it calls `\getanddefine@fonts` directly in future as well. We use the macro `\extract@alph@from@version` to do this. It takes the math alphabet identifier #1 and the math version macro as arguments.

```

622 % \expandafter\extract@alph@from@version
623 % \csname mv@\math@version\expandafter\endcsname
624 % \expandafter{\number\csname c@mv@\math@version\endcsname}%
625 % #1%
626 % \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.

```
627 %\expandafter #1\fi}
```

*(End of definition for `\select@group`.)*

#### `\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.

```
628 \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`.

```
629 \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`.

```
630 \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.

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

Then we execute our auxiliary macro.

```
632 \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.

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

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

```
634 \reserved@c{%
635 \getanddefine@fonts{#2}##2%
636 \install@mathalphabet#3{%
637 \relax\ifmmode \else \non@alpherr#3\fi
638 \use@mathgroup##1{#2}}}%
```

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

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

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

```
641 \expandafter\reserved@a\reserved@b\@nil
642 }
```

(*End of definition for `\extract@alph@from@version`.*)

`\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.

```
643 \let\math@bgroup\bgroup
644 \def\math@egroup#1{#1\egroup}
```

(*End of definition for `\math@bgroup` and `\math@egroup`.*)

`\calculate@math@sizes` Here is the default definition for `\calculate@math@sizes` a more elaborate interface is under testing in mthscale.sty.

```
645 \gdef\calculate@math@sizes{%
646 \@font@info{Calculating\space math\space sizes\space for\space
647 size\space <\f@size>}%
648 \dimen@\f@size \p@
649 \tempdima \scriptscriptfontratio \dimen@
650 \dimen@ \scriptfontratio \dimen@
651 \expandafter\xdef\csname S@\f@size\endcsname{%
652 \gdef\noexpand\font@size{\f@size}%
653 \gdef\noexpand\sf@size{\strip@pt\tempdima}%
654 \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
655 \noexpand\math@fonttrue}}
```

(*End of definition for `\calculate@math@sizes`.*)

\defaultscriptratio The default ratio for math sizes is:  
\defaultscriptscriptratio 1 to \defaultscriptratio to \defaultscriptscriptratio.  
By default this is 1 to .7 to .5.

```

656 \def\defaultscriptratio{.7}
657 \def\defaultscriptscriptratio{.5}

```

(End of definition for \defaultscriptratio and \defaultscriptscriptratio.)

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

```

658 \ifx\noaccents@\undefined
659 \let\noaccents@\empty
660 \fi

```

(End of definition for \noaccents@.)

\showhyphens The \showhyphens command must be redefined since the version in plain.tex uses \tenrm. We have also made some further adjustments for its use in L<sup>A</sup>T<sub>E</sub>X.

```

661 </2ekernel>
662 <latexrelease>\IncludeInRelease{2017/01/01}{\showhyphens}%
663 <latexrelease> {XeTeX support for \showhyphens}%
664 <*2ekernel | latexrelease>
665 \ifx\XeTeXcharclass\undefined

```

Version for engines other than XeT<sub>E</sub>X.

```

666 \DeclareRobustCommand\showhyphens[1]{%
667 \setbox0\vbox{%
668 \color@begingroup
669 \everypar{}%
670 \parfillskip\z@skip\hsize\maxdimen
671 \normalfont
672 \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\
673 \color@endgroup}}

```

```

674 \else

```

XeT<sub>E</sub>X version. When using system fonts XeT<sub>E</sub>X 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 would generate spurious Missing Character warnings in the log, these are however suppressed from the terminal and log output by ensuring that \tracinglostchars is locally zero.

```

675 \DeclareRobustCommand\showhyphens[1]{%
676 \setbox0\vbox{%
677 \usefont{TU}{lmr}{m}{n}%
678 \hsize 1sp %
679 \hbadness\@M
680 \hfuzz\maxdimen
681 \tracingonline\z@
682 \tracinglostchars\z@
683 \everypar{}%
684 \leftskip\z@skip
685 \rightskip\z@skip

```

```

686 \parfillskip\z@skip
687 \hyphenpenalty=-\@M
688 \pretolerance\m@ne
689 \interlinepenalty\z@
690 \clubpenalty\z@
691 \widowpenalty\z@
692 \brokenpenalty1127 %
693 \setbox\z@\hbox{}%
694 \noindent
695 \hskip\z@skip
696 #1%
697 \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`).

```

698 \loop
699 \@tempswafalse
700 \ifnum\lastnodetype=11\unskip\@tempswatrue\fi
701 \ifnum\lastnodetype=12\unkern\@tempswatrue\fi
702 \ifnum\lastnodetype=13 %
703 \count@\lastpenalty
704 \unpenalty\@tempswatrue
705 \fi
706 \ifnum\lastnodetype=\@ne
707 \setbox\tw@\lastbox\@tempswatrue
708 \setbox0\hbox{\unhbox\tw@\unskip\unskip\unpenalty
709 \ifnum\count@=1127 \else\ \fi
710 \unhbox0}%
711 \count@\z@
712 \fi
713 \if@tempswa
714 \repeat
715 \hbadness\z@
716 \hsize\maxdimen
717 \showboxdepth\z@
718 \tolerance\m@ne
719 \hyphenpenalty\z@
720 \noindent\unhbox\z@
721 } }
722 \fi
723 </2ekernel | latexrelease>
724 <latexrelease>\EndIncludeInRelease
725 <latexrelease>\IncludeInRelease{0000/00/00}{\showhyphens}%
726 <latexrelease> {XeTeX support for \showhyphens}%
727 <latexrelease>\gdef\showhyphens#1{%
728 <latexrelease> \setbox0\vbox{%
729 <latexrelease> \color@begingroup
730 <latexrelease> \everypar{}%
731 <latexrelease> \parfillskip\z@skip\hsize\maxdimen
732 <latexrelease> \normalfont
733 <latexrelease> \pretolerance\m@ne\tolerance\m@ne
734 <latexrelease> \hbadness\z@\showboxdepth\z@\ #1%
735 <latexrelease> \color@endgroup}%
736 <latexrelease>\EndIncludeInRelease

```

737 `(*2ekernel)`

(*End of definition for \showhyphens.*)

`\addto@hook` We need a macro to add tokens to a hook.

738 `\long\def\addto@hook#1#2{\#1\expandafter{\the#1#2}}`

(*End of definition for \addto@hook.*)

`\@vpt`

739 `\def\@vpt{5}`

(*End of definition for \@vpt.*)

`\@vipt`

740 `\def\@vipt{6}`

(*End of definition for \@vipt.*)

`\@viipt`

741 `\def\@viipt{7}`

(*End of definition for \@viipt.*)

`\@viiiip`

742 `\def\@viiiip{8}`

(*End of definition for \@viiiip.*)

`\@ixpt`

743 `\def\@ixpt{9}`

(*End of definition for \@ixpt.*)

`\@xpt`

744 `\def\@xpt{10}`

(*End of definition for \@xpt.*)

`\@xipt`

745 `\def\@xipt{10.95}`

(*End of definition for \@xipt.*)

`\@xiipt`

746 `\def\@xiipt{12}`

(*End of definition for \@xiipt.*)

`\@xivpt`

747 `\def\@xivpt{14.4}`

(*End of definition for \@xivpt.*)

`\@xviipt`

748 `\def\@xviipt{17.28}`

(End of definition for \@xviipt.)

\@xxpt

749 \def\@xxpt{20.74}

(End of definition for \@xxpt.)

\@xxvpt

750 \def\@xxvpt{24.88}

(End of definition for \@xxvpt.)

751 \langle/2ekernel\rangle

## File 25

# ltfssaxes.dtx

This file contains the implementation for handling extra axes splitting the series and the values into sub-categories. selection commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of the L<sup>A</sup>T<sub>E</sub>X Font Selection Scheme.

Everything in the this file got introduced 2020/02/02, so we use large rollback chunks, only interrupted if necessary.

```
1 <*2ekernel | latexrelease>
2 <latexrelease>\IncludeInRelease{2020/02/02}%
3 <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).

```
4 \def\DeclareFontSeriesChangeRule#1#2#3#4{%
5 @namedef{series@#1@#2}{\{\!\{#3\!\}{\!\!} {#4\!\}}}}
```

(End of definition for `\DeclareFontSeriesChangeRule`.)

## 1.2 Mapping rules for series changes

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

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

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

Design considerations for mapping entries:

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

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

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

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

- Also: while I did set up all nine standard weight values from `u1` 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.

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

```

18 \DeclareFontSeriesChangeRule {bx}{b}{bx}{}%
19 \DeclareFontSeriesChangeRule {bx}{c} {bc} {bx} %<-----
20 \DeclareFontSeriesChangeRule {bx}{eb}{ebx}{}%
21 \DeclareFontSeriesChangeRule {bx}{ec} {bec} {bx} %<-----
22 \DeclareFontSeriesChangeRule {bx}{el}{elx}{}%
23 \DeclareFontSeriesChangeRule {bx}{l}{lx}{}%
24 \DeclareFontSeriesChangeRule {bx}{sb} {sbx} {}%
25 \DeclareFontSeriesChangeRule {bx}{sc} {bsc} {bx} %<-----
26 \DeclareFontSeriesChangeRule {bx}{s1}{slx} {}%
27 \DeclareFontSeriesChangeRule {bx}{ub}{ubx}{}%
28 \DeclareFontSeriesChangeRule {bx}{ul}{ulx}{}%
29 \DeclareFontSeriesChangeRule {bx}{x}{bx}{}%

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

36 \DeclareFontSeriesChangeRule {c}{bx} {bx} {b} %<-----
37 \DeclareFontSeriesChangeRule {c}{b}{bc}{}%
38 \DeclareFontSeriesChangeRule {c}{eb}{ebc}{}%
39 \DeclareFontSeriesChangeRule {c}{el}{elc}{}%
40 \DeclareFontSeriesChangeRule {c}{l}{lc}{}%
41 \DeclareFontSeriesChangeRule {c}{sb}{sbc}{}%
42 \DeclareFontSeriesChangeRule {c}{s1}{slc}{}%
43 \DeclareFontSeriesChangeRule {c}{ub}{ubc}{}%
44 \DeclareFontSeriesChangeRule {c}{ul}{ulc}{}%
45 \DeclareFontSeriesChangeRule {c}{x}{x}{m} %<-----

46 \DeclareFontSeriesChangeRule {ebc}{b}{bc}{}%
47 \DeclareFontSeriesChangeRule {ebc}{c}{ebc}{}%
48 \DeclareFontSeriesChangeRule {ebc}{eb}{ebc}{}%
49 \DeclareFontSeriesChangeRule {ebc}{ec}{ebec}{ebc}%
50 \DeclareFontSeriesChangeRule {ebc}{el}{elc}{}%
51 \DeclareFontSeriesChangeRule {ebc}{l}{lc}{}%
52 \DeclareFontSeriesChangeRule {ebc}{sb}{sbc}{}%
53 \DeclareFontSeriesChangeRule {ebc}{sc}{ebsc}{ebc}%
54 \DeclareFontSeriesChangeRule {ebc}{s1}{slc}{}%
55 \DeclareFontSeriesChangeRule {ebc}{ub}{ubc}{}%
56 \DeclareFontSeriesChangeRule {ebc}{ul}{ulc}{}%
57 \DeclareFontSeriesChangeRule {ebc}{x}{ebx}{}%

58 \DeclareFontSeriesChangeRule {ec}{bx} {bx} {b} %<-----
59 \DeclareFontSeriesChangeRule {ec}{b}{bec}{}%
60 \DeclareFontSeriesChangeRule {ec}{eb}{ebc}{}%
61 \DeclareFontSeriesChangeRule {ec}{el}{elec}{}%
62 \DeclareFontSeriesChangeRule {ec}{l}{lec}{}%
63 \DeclareFontSeriesChangeRule {ec}{sb}{sbec}{}%
64 \DeclareFontSeriesChangeRule {ec}{s1}{sle}{}%
65 \DeclareFontSeriesChangeRule {ec}{ub}{ubec}{}%
66 \DeclareFontSeriesChangeRule {ec}{ul}{ulec}{}%
67 \DeclareFontSeriesChangeRule {ec}{x}{x}{m} %<-----

68 \DeclareFontSeriesChangeRule {sc}{bx} {bx} {b} %<-----
69 \DeclareFontSeriesChangeRule {sc}{b}{bsc}{}%

```

```

70 \DeclareFontSeriesChangeRule {sc}{eb}{ebsc}{}
71 \DeclareFontSeriesChangeRule {sc}{el}{elsc}{}
72 \DeclareFontSeriesChangeRule {sc}{l}{lsc}{}
73 \DeclareFontSeriesChangeRule {sc}{sb}{sbsc}{}
74 \DeclareFontSeriesChangeRule {sc}{s1}{slsc}{}
75 \DeclareFontSeriesChangeRule {sc}{ub}{ubsc}{}
76 \DeclareFontSeriesChangeRule {sc}{ul}{ulsc}{}
77 \DeclareFontSeriesChangeRule {sc}{x}{x}{m} %<-----

78 \DeclareFontSeriesChangeRule {ebx}{b}{bx}{}
79 \DeclareFontSeriesChangeRule {ebx}{c}{ebc}{}
80 \DeclareFontSeriesChangeRule {ebx}{eb}{ebx}{}
81 \DeclareFontSeriesChangeRule {ebx}{ec}{ebec}{}
82 \DeclareFontSeriesChangeRule {ebx}{el}{elx}{}
83 \DeclareFontSeriesChangeRule {ebx}{l}{lx}{}
84 \DeclareFontSeriesChangeRule {ebx}{sb}{sbx}{}
85 \DeclareFontSeriesChangeRule {ebx}{sc}{ebsc}{}
86 \DeclareFontSeriesChangeRule {ebx}{s1}{slx}{}
87 \DeclareFontSeriesChangeRule {ebx}{ub}{ubx}{}
88 \DeclareFontSeriesChangeRule {ebx}{ul}{ulx}{}
89 \DeclareFontSeriesChangeRule {ebx}{x}{ebx}{}

90 \DeclareFontSeriesChangeRule {eb}{c}{ebc}{}
91 \DeclareFontSeriesChangeRule {eb}{ec}{ebec}{}
92 \DeclareFontSeriesChangeRule {eb}{sc}{ebsc}{}
93 \DeclareFontSeriesChangeRule {eb}{x}{ebx}{}

94 \DeclareFontSeriesChangeRule {elc}{b}{bc}{}
95 \DeclareFontSeriesChangeRule {elc}{c}{elc}{}
96 \DeclareFontSeriesChangeRule {elc}{eb}{ebc}{}
97 \DeclareFontSeriesChangeRule {elc}{ec}{elec}{}
98 \DeclareFontSeriesChangeRule {elc}{el}{elc}{}
99 \DeclareFontSeriesChangeRule {elc}{l}{lc}{}
100 \DeclareFontSeriesChangeRule {elc}{sb}{sbc}{}
101 \DeclareFontSeriesChangeRule {elc}{sc}{elsc}{}
102 \DeclareFontSeriesChangeRule {elc}{s1}{slc}{}
103 \DeclareFontSeriesChangeRule {elc}{ub}{ubc}{}
104 \DeclareFontSeriesChangeRule {elc}{ul}{ulc}{}
105 \DeclareFontSeriesChangeRule {elc}{x}{elx}{}

106 \DeclareFontSeriesChangeRule {elx}{b}{bx}{}
107 \DeclareFontSeriesChangeRule {elx}{c}{elc}{}
108 \DeclareFontSeriesChangeRule {elx}{eb}{ebx}{}
109 \DeclareFontSeriesChangeRule {elx}{ec}{elec}{}
110 \DeclareFontSeriesChangeRule {elx}{el}{elx}{}
111 \DeclareFontSeriesChangeRule {elx}{l}{lx}{}
112 \DeclareFontSeriesChangeRule {elx}{sb}{sbx}{}
113 \DeclareFontSeriesChangeRule {elx}{sc}{elsc}{}
114 \DeclareFontSeriesChangeRule {elx}{s1}{slx}{}
115 \DeclareFontSeriesChangeRule {elx}{ub}{ubx}{}
116 \DeclareFontSeriesChangeRule {elx}{ul}{ulx}{}
117 \DeclareFontSeriesChangeRule {elx}{x}{elx}{}

118 \DeclareFontSeriesChangeRule {el}{c}{elc}{}
119 \DeclareFontSeriesChangeRule {el}{ec}{elec}{}
120 \DeclareFontSeriesChangeRule {el}{sc}{elsc}{}
121 \DeclareFontSeriesChangeRule {el}{x}{elx}{}

```

```

122 \DeclareFontSeriesChangeRule {lc}{b}{bc}={}
123 \DeclareFontSeriesChangeRule {lc}{c}{lc}={}
124 \DeclareFontSeriesChangeRule {lc}{eb}{ebc}={}
125 \DeclareFontSeriesChangeRule {lc}{ec}{lec}={}
126 \DeclareFontSeriesChangeRule {lc}{el}{elc}={}
127 \DeclareFontSeriesChangeRule {lc}{l}{lc}={}
128 \DeclareFontSeriesChangeRule {lc}{sb}{sbc}={}
129 \DeclareFontSeriesChangeRule {lc}{sc}{lsc}={}
130 \DeclareFontSeriesChangeRule {lc}{s1}{slc}={}
131 \DeclareFontSeriesChangeRule {lc}{ub}{ubc}={}
132 \DeclareFontSeriesChangeRule {lc}{ul}{ulc}={}
133 \DeclareFontSeriesChangeRule {lc}{x}{lx}={}

134 \DeclareFontSeriesChangeRule {lx}{b}{bx}={}
135 \DeclareFontSeriesChangeRule {lx}{c}{lc}={}
136 \DeclareFontSeriesChangeRule {lx}{eb}{ebx}={}
137 \DeclareFontSeriesChangeRule {lx}{ec}{lec}={}
138 \DeclareFontSeriesChangeRule {lx}{el}{elx}={}
139 \DeclareFontSeriesChangeRule {lx}{l}{lx}={}
140 \DeclareFontSeriesChangeRule {lx}{sb}{sbx}={}
141 \DeclareFontSeriesChangeRule {lx}{sc}{lsc}={}
142 \DeclareFontSeriesChangeRule {lx}{s1}{slx}={}
143 \DeclareFontSeriesChangeRule {lx}{ub}{ubx}={}
144 \DeclareFontSeriesChangeRule {lx}{ul}{ulx}={}
145 \DeclareFontSeriesChangeRule {lx}{x}{lx}={}

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

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

161 \DeclareFontSeriesChangeRule {sbc}{b}{bc}={}
162 \DeclareFontSeriesChangeRule {sbc}{c}{sbc}={}
163 \DeclareFontSeriesChangeRule {sbc}{eb}{ebc}={}
164 \DeclareFontSeriesChangeRule {sbc}{ec}{sbec}{sbc}
165 \DeclareFontSeriesChangeRule {sbc}{el}{elc}={}
166 \DeclareFontSeriesChangeRule {sbc}{l}{lc}={}
167 \DeclareFontSeriesChangeRule {sbc}{sb}{sbc}={}
168 \DeclareFontSeriesChangeRule {sbc}{sc}{sbsc}{sbc}
169 \DeclareFontSeriesChangeRule {sbc}{s1}{slc}={}
170 \DeclareFontSeriesChangeRule {sbc}{ub}{ubc}={}
171 \DeclareFontSeriesChangeRule {sbc}{ul}{ulc}={}
172 \DeclareFontSeriesChangeRule {sbc}{x}{sbx}={}

173 \DeclareFontSeriesChangeRule {sbx}{b}{bx}={}

```

```

174 \DeclareFontSeriesChangeRule {sbx}{c}{sbc}{}

175 \DeclareFontSeriesChangeRule {sbx}{eb}{ebx}{}

176 \DeclareFontSeriesChangeRule {sbx}{ec}{sbec}{}

177 \DeclareFontSeriesChangeRule {sbx}{el}{elx}{}

178 \DeclareFontSeriesChangeRule {sbx}{l}{lx}{}

179 \DeclareFontSeriesChangeRule {sbx}{sb}{sbx}{}

180 \DeclareFontSeriesChangeRule {sbx}{sc}{sbsc}{}

181 \DeclareFontSeriesChangeRule {sbx}{s1}{slx}{}

182 \DeclareFontSeriesChangeRule {sbx}{ub}{ubx}{}

183 \DeclareFontSeriesChangeRule {sbx}{ul}{ulx}{}

184 \DeclareFontSeriesChangeRule {sbx}{x}{sbx}{}

185 \DeclareFontSeriesChangeRule {sb}{c} {sbc} {bc} %? %<----

186 \DeclareFontSeriesChangeRule {sb}{ec} {sbec} {sbc} %? %<----

187 \DeclareFontSeriesChangeRule {sb}{sc} {sbsc} {sbc} %? %<----

188 \DeclareFontSeriesChangeRule {sb}{x} {sbx} {bx} %? %<----

189 \DeclareFontSeriesChangeRule {slc}{b}{bc}{}

190 \DeclareFontSeriesChangeRule {slc}{c}{slc}{}

191 \DeclareFontSeriesChangeRule {slc}{eb}{ebc}{}

192 \DeclareFontSeriesChangeRule {slc}{ec}{slec}{}

193 \DeclareFontSeriesChangeRule {slc}{el}{elc}{}

194 \DeclareFontSeriesChangeRule {slc}{l}{lc}{}

195 \DeclareFontSeriesChangeRule {slc}{sb}{sbc}{}

196 \DeclareFontSeriesChangeRule {slc}{sc}{slsc}{}

197 \DeclareFontSeriesChangeRule {slc}{s1}{slc}{}

198 \DeclareFontSeriesChangeRule {slc}{ub}{ubc}{}

199 \DeclareFontSeriesChangeRule {slc}{ul}{ulc}{}

200 \DeclareFontSeriesChangeRule {slc}{x}{slx}{}

201 \DeclareFontSeriesChangeRule {slx}{b}{bx}{}

202 \DeclareFontSeriesChangeRule {slx}{c}{slc}{}

203 \DeclareFontSeriesChangeRule {slx}{eb}{ebx}{}

204 \DeclareFontSeriesChangeRule {slx}{ec}{slec}{}

205 \DeclareFontSeriesChangeRule {slx}{el}{elx}{}

206 \DeclareFontSeriesChangeRule {slx}{l}{lx}{}

207 \DeclareFontSeriesChangeRule {slx}{sb}{sbx}{}

208 \DeclareFontSeriesChangeRule {slx}{sc}{slsc}{}

209 \DeclareFontSeriesChangeRule {slx}{s1}{slx}{}

210 \DeclareFontSeriesChangeRule {slx}{ub}{ubx}{}

211 \DeclareFontSeriesChangeRule {slx}{ul}{ulx}{}

212 \DeclareFontSeriesChangeRule {slx}{x}{slx}{}

213 \DeclareFontSeriesChangeRule {s1}{c}{slc}{}

214 \DeclareFontSeriesChangeRule {s1}{ec}{slec}{}

215 \DeclareFontSeriesChangeRule {s1}{sc}{slsc}{}

216 \DeclareFontSeriesChangeRule {s1}{x}{slx}{}

217 \DeclareFontSeriesChangeRule {ubc}{b}{bc}{}

218 \DeclareFontSeriesChangeRule {ubc}{c}{ubc}{}

219 \DeclareFontSeriesChangeRule {ubc}{eb}{ebc}{}

220 \DeclareFontSeriesChangeRule {ubc}{ec}{ubec}{}

221 \DeclareFontSeriesChangeRule {ubc}{el}{elc}{}

222 \DeclareFontSeriesChangeRule {ubc}{l}{lc}{}

223 \DeclareFontSeriesChangeRule {ubc}{sb}{sbc}{}

224 \DeclareFontSeriesChangeRule {ubc}{sc}{ubsc}{}

225 \DeclareFontSeriesChangeRule {ubc}{s1}{slc}{}


```

```

226 \DeclareFontSeriesChangeRule {ubc}{ub}{ubc}{}
227 \DeclareFontSeriesChangeRule {ubc}{ul}{ulc}{}
228 \DeclareFontSeriesChangeRule {ubc}{x}{ubx}{}
229 \DeclareFontSeriesChangeRule {ubx}{b}{bx}{}
230 \DeclareFontSeriesChangeRule {ubx}{c}{ubc}{}
231 \DeclareFontSeriesChangeRule {ubx}{eb}{ebx}{}
232 \DeclareFontSeriesChangeRule {ubx}{ec}{ubec}{}
233 \DeclareFontSeriesChangeRule {ubx}{el}{elx}{}
234 \DeclareFontSeriesChangeRule {ubx}{l}{lx}{}
235 \DeclareFontSeriesChangeRule {ubx}{sb}{sbx}{}
236 \DeclareFontSeriesChangeRule {ubx}{sc}{ubsc}{}
237 \DeclareFontSeriesChangeRule {ubx}{s1}{slx}{}
238 \DeclareFontSeriesChangeRule {ubx}{ub}{ubx}{}
239 \DeclareFontSeriesChangeRule {ubx}{ul}{ulx}{}
240 \DeclareFontSeriesChangeRule {ubx}{x}{ubx}{}
241 \DeclareFontSeriesChangeRule {ub}{c}{ubc}{}
242 \DeclareFontSeriesChangeRule {ub}{ec}{ubec}{}
243 \DeclareFontSeriesChangeRule {ub}{sc}{ubsc}{}
244 \DeclareFontSeriesChangeRule {ub}{x}{ubx}{}
245 \DeclareFontSeriesChangeRule {ulc}{b}{bc}{}
246 \DeclareFontSeriesChangeRule {ulc}{c}{ulc}{}
247 \DeclareFontSeriesChangeRule {ulc}{eb}{ebc}{}
248 \DeclareFontSeriesChangeRule {ulc}{ec}{ulec}{ulc}
249 \DeclareFontSeriesChangeRule {ulc}{el}{elc}{}
250 \DeclareFontSeriesChangeRule {ulc}{l}{lc}{}
251 \DeclareFontSeriesChangeRule {ulc}{sb}{sbc}{}
252 \DeclareFontSeriesChangeRule {ulc}{sc}{ulsc}{ulc}
253 \DeclareFontSeriesChangeRule {ulc}{s1}{slc}{}
254 \DeclareFontSeriesChangeRule {ulc}{ub}{ubc}{}
255 \DeclareFontSeriesChangeRule {ulc}{ul}{ulc}{}
256 \DeclareFontSeriesChangeRule {ulc}{x}{ulx}{}
257 \DeclareFontSeriesChangeRule {ulx}{b}{bx}{}
258 \DeclareFontSeriesChangeRule {ulx}{c}{ulc}{}
259 \DeclareFontSeriesChangeRule {ulx}{eb}{ebx}{}
260 \DeclareFontSeriesChangeRule {ulx}{ec}{ulec}{}
261 \DeclareFontSeriesChangeRule {ulx}{el}{elx}{}
262 \DeclareFontSeriesChangeRule {ulx}{l}{lx}{}
263 \DeclareFontSeriesChangeRule {ulx}{sb}{sbx}{}
264 \DeclareFontSeriesChangeRule {ulx}{sc}{ulsc}{}
265 \DeclareFontSeriesChangeRule {ulx}{s1}{slx}{}
266 \DeclareFontSeriesChangeRule {ulx}{ub}{ubx}{}
267 \DeclareFontSeriesChangeRule {ulx}{ul}{ulx}{}
268 \DeclareFontSeriesChangeRule {ulx}{x}{ulx}{}
269 \DeclareFontSeriesChangeRule {ul}{c}{ulc}{}
270 \DeclareFontSeriesChangeRule {ul}{ec}{ulec}{}
271 \DeclareFontSeriesChangeRule {ul}{sc}{ulsc}{}
272 \DeclareFontSeriesChangeRule {ul}{x}{ulx}{}
273 \DeclareFontSeriesChangeRule {x}{b}{bx}{}
274 \DeclareFontSeriesChangeRule {x}{c}{c}{}
275 \DeclareFontSeriesChangeRule {x}{eb}{ebx}{}
276 \DeclareFontSeriesChangeRule {x}{ec}{ec}{}
277 \DeclareFontSeriesChangeRule {x}{el}{elx}{}

```

```

278 \DeclareFontSeriesChangeRule {x}{l}{lx}{}
279 \DeclareFontSeriesChangeRule {x}{sb}{sbx}{}
280 \DeclareFontSeriesChangeRule {x}{sc}{sc}{}
281 \DeclareFontSeriesChangeRule {x}{sl}{slx}{}
282 \DeclareFontSeriesChangeRule {x}{ub}{ubx}{}
283 \DeclareFontSeriesChangeRule {x}{ul}{ulx}{}

```

Special rules for `1m` etc. aren't needed because if the target `1m` is requested it will be used if there is no rule and that is then reduced to `1` automatically. Same for `mc` and friends. Only `?m` and `m?` need rules.

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

```

284 \DeclareFontSeriesChangeRule {bc}{m?}{c}{}
285 \DeclareFontSeriesChangeRule {bec}{m?}{ec}{}
286 \DeclareFontSeriesChangeRule {bsc}{m?}{sc}{}
287 \DeclareFontSeriesChangeRule {bx}{m?}{x}{}
288 \DeclareFontSeriesChangeRule {b}{m?}{m}{}
289 \DeclareFontSeriesChangeRule {c}{m?}{c}{}
290 \DeclareFontSeriesChangeRule {ebc}{m?}{c}{}
291 \DeclareFontSeriesChangeRule {ebec}{m?}{ec}{}
292 \DeclareFontSeriesChangeRule {ebsc}{m?}{sc}{}
293 \DeclareFontSeriesChangeRule {ebx}{m?}{x}{}
294 \DeclareFontSeriesChangeRule {eb}{m?}{m}{}
295 \DeclareFontSeriesChangeRule {ec}{m?}{ec}{}
296 \DeclareFontSeriesChangeRule {elc}{m?}{c}{}
297 \DeclareFontSeriesChangeRule {elec}{m?}{ec}{}
298 \DeclareFontSeriesChangeRule {elsc}{m?}{sc}{}
299 \DeclareFontSeriesChangeRule {elx}{m?}{x}{}
300 \DeclareFontSeriesChangeRule {el}{m?}{m}{}
301 \DeclareFontSeriesChangeRule {lc}{m?}{c}{}
302 \DeclareFontSeriesChangeRule {lec}{m?}{ec}{}
303 \DeclareFontSeriesChangeRule {lsc}{m?}{sc}{}
304 \DeclareFontSeriesChangeRule {lx}{m?}{x}{}
305 \DeclareFontSeriesChangeRule {l}{m?}{m}{}
306 \DeclareFontSeriesChangeRule {m}{m?}{m}{}
307 \DeclareFontSeriesChangeRule {sbc}{m?}{c}{}
308 \DeclareFontSeriesChangeRule {sbec}{m?}{ec}{}
309 \DeclareFontSeriesChangeRule {sbsc}{m?}{sc}{}
310 \DeclareFontSeriesChangeRule {sbx}{m?}{x}{}
311 \DeclareFontSeriesChangeRule {sb}{m?}{m}{}
312 \DeclareFontSeriesChangeRule {sc}{m?}{sc}{}
313 \DeclareFontSeriesChangeRule {slc}{m?}{c}{}
314 \DeclareFontSeriesChangeRule {slec}{m?}{ec}{}
315 \DeclareFontSeriesChangeRule {slsc}{m?}{sc}{}
316 \DeclareFontSeriesChangeRule {slx}{m?}{x}{}
317 \DeclareFontSeriesChangeRule {s1}{m?}{m}{}
318 \DeclareFontSeriesChangeRule {ubc}{m?}{c}{}
319 \DeclareFontSeriesChangeRule {ubec}{m?}{ec}{}
320 \DeclareFontSeriesChangeRule {ubsc}{m?}{sc}{}
321 \DeclareFontSeriesChangeRule {ubx}{m?}{x}{}
322 \DeclareFontSeriesChangeRule {ub}{m?}{ub}{}
323 \DeclareFontSeriesChangeRule {ulc}{m?}{c}{}
324 \DeclareFontSeriesChangeRule {ulec}{m?}{ec}{}
325 \DeclareFontSeriesChangeRule {ulsc}{m?}{sc}{}
326 \DeclareFontSeriesChangeRule {ulx}{m?}{x}{}

```

```

327 \DeclareFontSeriesChangeRule {ul}{m?}{m}{}
328 \DeclareFontSeriesChangeRule {x}{m?}{x}{}

 And there the special rules for ?m:
329 \DeclareFontSeriesChangeRule {bc}{?m}{b}{}
330 \DeclareFontSeriesChangeRule {bec}{?m}{b}{}
331 \DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
332 \DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
333 \DeclareFontSeriesChangeRule {bx}{?m}{b}{}
334 \DeclareFontSeriesChangeRule {b}{?m}{b}{}
335 \DeclareFontSeriesChangeRule {c}{?m}{m}{}
336 \DeclareFontSeriesChangeRule {ebc}{?m}{eb}{}
337 \DeclareFontSeriesChangeRule {ebec}{?m}{eb}{}
338 \DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
339 \DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
340 \DeclareFontSeriesChangeRule {ebx}{?m}{eb}{}
341 \DeclareFontSeriesChangeRule {eb}{?m}{eb}{}
342 \DeclareFontSeriesChangeRule {ec}{?m}{m}{}
343 \DeclareFontSeriesChangeRule {elc}{?m}{el}{}
344 \DeclareFontSeriesChangeRule {elec}{?m}{el}{}
345 \DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
346 \DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
347 \DeclareFontSeriesChangeRule {elx}{?m}{el}{}
348 \DeclareFontSeriesChangeRule {el}{?m}{el}{}
349 \DeclareFontSeriesChangeRule {lc}{?m}{l}{}
350 \DeclareFontSeriesChangeRule {lec}{?m}{l}{}
351 \DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
352 \DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
353 \DeclareFontSeriesChangeRule {lx}{?m}{l}{}
354 \DeclareFontSeriesChangeRule {l}{?m}{l}{}
355 \DeclareFontSeriesChangeRule {m}{?m}{m}{}
356 \DeclareFontSeriesChangeRule {sbc}{?m}{sb}{}
357 \DeclareFontSeriesChangeRule {sbec}{?m}{sb}{}
358 \DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
359 \DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
360 \DeclareFontSeriesChangeRule {sbx}{?m}{sb}{}
361 \DeclareFontSeriesChangeRule {sb}{?m}{sb}{}
362 \DeclareFontSeriesChangeRule {sc}{?m}{m}{}
363 \DeclareFontSeriesChangeRule {sc}{?m}{m}{}
364 \DeclareFontSeriesChangeRule {slc}{?m}{s1}{}
365 \DeclareFontSeriesChangeRule {slec}{?m}{s1}{}
366 \DeclareFontSeriesChangeRule {slsc}{?m}{s1}{}
367 \DeclareFontSeriesChangeRule {slsc}{?m}{s1}{}
368 \DeclareFontSeriesChangeRule {slx}{?m}{s1}{}
369 \DeclareFontSeriesChangeRule {sl}{?m}{s1}{}
370 \DeclareFontSeriesChangeRule {ubc}{?m}{ub}{}
371 \DeclareFontSeriesChangeRule {ubec}{?m}{ub}{}
372 \DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
373 \DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
374 \DeclareFontSeriesChangeRule {ubx}{?m}{ub}{}
375 \DeclareFontSeriesChangeRule {ub}{?m}{m}{}
376 \DeclareFontSeriesChangeRule {ulc}{?m}{ul}{}
377 \DeclareFontSeriesChangeRule {ulec}{?m}{ul}{}
378 \DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}
379 \DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}

```

```

380 \DeclareFontSeriesChangeRule {ulx}{?m}{ul}{}
381 \DeclareFontSeriesChangeRule {ul}{?m}{ul}{}
382 \DeclareFontSeriesChangeRule {x}{?m}{m}{}

 Supporting rollback ...

383 </2ekernel | latexrelease>
384 <latexrelease>\EndIncludeInRelease
385 <latexrelease>\IncludeInRelease{0000/00/00}%
386 <latexrelease> {\DeclareFontSeriesChangeRule}{Series change rules}%
387 <latexrelease>
388 <latexrelease>\let\DeclareFontSeriesChangeRule\@undefined
389 <latexrelease>
390 <latexrelease>\EndIncludeInRelease

```

### 1.3 Changing to a new series

```

391 {*2ekernel | latexrelease}
392 <latexrelease>\IncludeInRelease{2021/06/01}%
393 <latexrelease> {\fontseries}{delay fontseries update}%

```

**\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`. But the lookup should be done within the target family and call to `\fontseries` might be followed by a `\fontfamily` call. So we delay the processing to `\selectfont` and only record the necessary action in `\delayed@f@adjustment`.

```

394 \DeclareRobustCommand\fontseries[1]{\@forced@seriesfalse
395 \expandafter\def\expandafter\delayed@f@adjustment\expandafter
396 {\delayed@f@adjustment\delayed@merge@font@series{\#1}}}

```

(*End of definition for \fontseries.*)

**\delayed@f@adjustment** The macro holding the delayed action(s) for use in `\selectfont`.

```

397 \let\delayed@f@adjustment\empty

```

(*End of definition for \delayed@f@adjustment.*)

**\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.

```

398 \DeclareRobustCommand\fontseriesforce[1]{\@forced@seriestrue
399 \expandafter\def\expandafter\delayed@f@adjustment\expandafter
400 {\delayed@f@adjustment\edef\f@series{\#1}}}

```

(*End of definition for \fontseriesforce.*)

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

```

401 \newif\if@forced@series

```

(*End of definition for \if@forced@series.*)

```

402 </2ekernel | latexrelease>
403 <latexrelease>\EndIncludeInRelease

```

```

404 〈latexrelease〉\IncludeInRelease{2020/02/02}%
405 〈latexrelease〉 {\fontseries}{delay fontseries update}%
406 〈latexrelease〉
407 〈latexrelease〉\DeclareRobustCommand\fontseries[1]{\@forced@seriesfalse
408 〈latexrelease〉 \merge@font@series{\#1}%
409 〈latexrelease〉\DeclareRobustCommand\fontseriesforce[1]{\@forced@seriestrue
410 〈latexrelease〉 \edef\f@series{\#1}%
411 〈latexrelease〉\let\delayed@f@adjustment\@undefined
412 〈latexrelease〉

```

For a roll forward we may have to define `\if@forced@series` but this needs doing in a way that `\TeX` doesn't see it when skipping over conditionals.

```

413 〈latexrelease〉\expandafter\newif\csname if@forced@series\endcsname
414 〈latexrelease〉
415 〈latexrelease〉\EndIncludeInRelease

416 〈latexrelease〉\IncludeInRelease{0000/00/00}%
417 〈latexrelease〉 {\fontseries}{delay fontseries update}%
418 〈latexrelease〉
419 〈latexrelease〉\DeclareRobustCommand\fontseries[1]{\edef\f@series{\#1}}
420 〈latexrelease〉\let\fontseriesforce\@undefined
421 〈latexrelease〉
422 〈latexrelease〉\EndIncludeInRelease

423 〈*2ekernel | latexrelease〉
424 〈latexrelease〉\IncludeInRelease{2020/02/02}%
425 〈latexrelease〉 {\merge@font@series}{Merge series values}%

```

`\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.

```

426 \def\merge@font@series#1{%
427 \expandafter\expandafter\expandafter
428 \merge@font@series@
429 \csname series@\f@series \#1\endcsname
430 {\#1}%
431 \nil
432 }

```

(End of definition for `\merge@font@series`.)

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

```

433 \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.

```

434 \%ifcat\expandafter X\detokenize{\#1}X%
435 \def\reserved@a{\#3}%
436 \ifx\reserved@a\empty
437 \set@target@series{\#2}%
438 \else

```

Otherwise we check if the desired result for the series (#1) exists for the font family and the current shape. All this happens inside `\selectfont` which has already taken care to load the `.fd` file if necessary.

```
439 \edef\reserved@a{\f@encoding / \f@family / #1 / \f@shape}%
440 \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).

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

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

```
442 \else
443 \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.

```
444 \set@target@series{#2}%
445 \font@shape@subst@warning
```

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

```
446 \else
447 \set@target@series{#3}%
448 \font@shape@subst@warning
449 \fi
450 \fi
451 \fi
452 }
```

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 `\@reserved@a` above instead of doing inline testing inside `\ifcsname`).

```
453 \def\font@shape@subst@warning{%
454 \edef\reserved@b{\curr@fontshape}%
455 \ifx\reserved@a\reserved@b \else
456 \font@warning{Font shape '\reserved@a' undefined\MessageBreak
457 using '\reserved@b' instead}%
458 \fi
459 }
```

*(End of definition for `\merge@font@series`.)*

```
\merge@font@series@without@substitution
\merge@font@series@without@substitution@
\delayed@merge@font@series
```

`\merge@font@series@without@substitution` works like `\merge@font@series`, i.e., it looks up the combination in the rule base and if there exists an entry it uses it and if not it uses the new series value. However, it doesn't check if there is actually a font face with the new series value as `\merge@font@series` does. This simplified command is used in `\selectfont` at a point where other font attributes are not yet updated so that checking the font face might result incorrect in substitutions.

```
460 \def\merge@font@series@without@substitution#1{%
461 \expandafter\expandafter\expandafter
462 \merge@font@series@without@substitution@
463 \csname series@\f@series @#1\endcsname
```

```

464 {#1}%
465 \@nil
466 }
467 \def\merge@font@series@without@substitution{\@nil{%
468 \def\reserved@a{#3}%
469 \ifx\reserved@a\empty
470 \set@target@series{#2}%
471 \else
472 \set@target@series{#1}%
473 \fi
474 }

(End of definition for \merge@font@series@without@substitution,
 \merge@font@series@without@substitution@, and \delayed@merge@font@series.)

```

\delayed@merge@font@series When we delay the merge action in \fontseries we first attempt to use merging without substitution. If that results in a non-existing font face the merge is redone in \selectfont using a version with substitution. See \selectfont for details.

```
475 \let\delayed@merge@font@series\merge@font@series@without@substitution
```

(End of definition for \delayed@merge@font@series.)

\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 and we also want to change \typeout so that it doesn't report missing .fd files on the terminal).

```

476 \def\maybe@load@fontshape{%
477 \begingroup
478 \let \typeout \font@info
479 \try@load@fontshape
480 \endgroup

```

(End of definition for \maybe@load@fontshape.)

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

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

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

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

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

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

```

482 \edef\f@series{#1}%
483 \series@maybe@drop@one@m\f@series\f@series
484 }
```

(End of definition for \set@target@series.)

\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 its 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.

```
485 \def\series@maybe@drop@one@m#1{%
486 \expandafter\series@maybe@drop@one@m@\expandafter{\#1}}%
487
488 \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`.

```
489 \def\in@##1,#1,{}%
490 \series@check@toks\expandafter{\in@%
491 ,ulm,elm,lm,slm,mm,sbm,bm,ebm,ubm,muc,mec,mc,msc,msx,mx,mex,mux,{}{},#1,}%
492 \edef\in@{\the\series@check@toks}%
493 \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.

```
494 \edef#2{\#1}%
495 \else
496 \edef#2{\expandafter\series@drop@one@m #1\series@drop@one@m}%
497 \fi
498 }
```

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

```
499 \newtoks\series@check@toks
```

*(End of definition for `\series@maybe@drop@one@m`.)*

\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.

```
500 \def\series@drop@one@m#1m#2m#3\series@drop@one@m{%
501 % \ifx\relax#1#2\relax m\else#1#2\fi
502 #1#2%
503 }
```

*(End of definition for `\series@drop@one@m`.)*

Supporting rollback ...

```
504 </2ekernel | latexrelease>
505 <latexrelease>\EndIncludeInRelease
506 <latexrelease>\IncludeInRelease{0000/00/00}%
507 <latexrelease> {\merge@font@series}{Merge series values}%
508 <latexrelease>
509 <latexrelease>\let\merge@font@series@\undefined
510 <latexrelease>\let\merge@font@series@\@undefined
511 <latexrelease>\let@font@shape@subst@warning@\undefined
512 <latexrelease>\let\merge@font@series@without@substitution@\undefined
513 <latexrelease>\let\merge@font@series@without@substitution@\@undefined
```

```

514 〈latexrelease〉\let\delayed@merge@font@series\@undefined
515 〈latexrelease〉\let\maybe@load@fontshape\@undefined
516 〈latexrelease〉\let\set@target@series\@undefined
517 〈latexrelease〉\let\series@maybe@drop@one@m\@undefined
518 〈latexrelease〉\let\series@drop@one@m\@undefined
519 〈latexrelease〉
520 〈latexrelease〉\EndIncludeInRelease

```

## 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.

```

521 {*2ekernel | latexrelease}
522 〈latexrelease〉\IncludeInRelease{2020/02/02}%
523 〈latexrelease〉 {\DeclareFontShapeChangeRule}{Font shape change rules}%

```

`\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).

```

524 \def\DeclareFontShapeChangeRule #1#2#3#4{%
525 @namedef{shape@#1@#2}{\{\#3\}\{\#4\}}}

```

(*End of definition for \DeclareFontShapeChangeRule.*)

There is kind of the same problem with returning back from `sc` to normal. It sort of needs its own letter. In `fontspec` this was solved by the first time `\upshape` changes `it` or `sl` back (so only `sc` remains) and second time it changes then `sc` back to normal. Maybe that’s not a bad way to handle it, but decided for a slightly different approach: `n` always returns to “normal”, ie resets everything and `up` changes italic or slanted to upright and `ulc` undoes small caps.

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

`\ulcshape` To request going back to upper/lowercase we need a new command. It uses `ulc` 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).

```

526 \DeclareRobustCommand\ulcshape
527 {\not@math@alphabet\ulcshape\relax
528 \fontshape\ulcdefault\selectfont}
529 \let\ulcdefault\@undefined % for rollback
530 \newcommand\ulcdefault{ulc}

```

(*End of definition for \ulcshape , \textulc , and \ulcdefault.*)

`\swshape` 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.

```

531 \DeclareRobustCommand\swshape
532 {\not@math@alphabet\swshape\relax
533 \fontshape\swdefault\selectfont}
534 \let\swdefault\@undefined % for rollback
535 \newcommand\swdefault{sw}

```

(End of definition for `\swshape`, `\textsw`, and `\swdefault`.)

`\sscshape` 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`.

```
536 \DeclareRobustCommand\sscshape
537 {\not@math@alphabet\sscshape\relax
538 \fontshape\sscdefault\selectfont}
539 \let\sscdefault\@undefined % for rollback
540 \newcommand\sscdefault{sc}
```

(End of definition for `\sscshape`, `\textssc`, and `\sscdefault`.)

## 2.1 Mapping rules for shape combinations

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

```
541 \%{\DeclareFontShapeChangeRule {n}{n} {n} {}}
542 \DeclareFontShapeChangeRule {n}{it} {it} {sl}
543 \DeclareFontShapeChangeRule {n}{sl} {sl} {it}
544 \%{\DeclareFontShapeChangeRule {n}{sw} {sw} {}}
545 \%{\DeclareFontShapeChangeRule {n}{sc} {sc} {}}
546 \DeclareFontShapeChangeRule {n}{ulc} {n} {}
547 \DeclareFontShapeChangeRule {n}{up} {n} {}
548 \%{\DeclareFontShapeChangeRule {it}{n} {n} {}}
549 \%{\DeclareFontShapeChangeRule {it}{it} {it} {}}
550 \DeclareFontShapeChangeRule {it}{sl} {sl} {it}
551 \%{\DeclareFontShapeChangeRule {it}{sw} {sw} {}}
```

If neither `scit` nor `scls` 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.

```
552 \DeclareFontShapeChangeRule {it}{sc} {scit} {scls}
553 \DeclareFontShapeChangeRule {it}{ulc} {it} {}
554 \DeclareFontShapeChangeRule {it}{up} {n} {}
555 \%{\DeclareFontShapeChangeRule {sl}{n} {n} {}}
556 \DeclareFontShapeChangeRule {sl}{it} {it} {sl}
557 \%{\DeclareFontShapeChangeRule {sl}{sl} {sl} {}}
558 \%{\DeclareFontShapeChangeRule {sl}{sw} {sw} {}}
559 \DeclareFontShapeChangeRule {sl}{sc} {scls} {scit}
560 \DeclareFontShapeChangeRule {sl}{ulc} {sl} {}
561 \DeclareFontShapeChangeRule {sl}{up} {n} {}
562 \%{\DeclareFontShapeChangeRule {sc}{n} {n} {}}
563 \DeclareFontShapeChangeRule {sc}{it} {scit} {scls}
564 \DeclareFontShapeChangeRule {sc}{sl} {scls} {scit}
565 \DeclareFontShapeChangeRule {sc}{sw} {scsw} {sw}
566 \%{\DeclareFontShapeChangeRule {sc}{sc} {sc} {}}
567 \DeclareFontShapeChangeRule {sc}{ulc} {n} {}
```

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

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

```

568 %\DeclareFontShapeChangeRule {sc}{up} {sc} {}
569 \DeclareFontShapeChangeRule {sc}{up} {n} {}
570 %\DeclareFontShapeChangeRule {scit}{n} {n} {}
571 \DeclareFontShapeChangeRule {scit}{it} {scit} {}
572 \DeclareFontShapeChangeRule {scit}{sl} {scls} {scit}
573 \DeclareFontShapeChangeRule {scit}{sw} {scsw} {sc} % or scit?
574 \DeclareFontShapeChangeRule {scit}{sc} {scit} {}
575 \DeclareFontShapeChangeRule {scit}{ulc} {it} {}
576 \DeclareFontShapeChangeRule {scit}{up} {sc} {}

```

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

```

577 %\DeclareFontShapeChangeRule {scls}{n} {n} {}
578 \DeclareFontShapeChangeRule {scls}{it} {scit} {scls}
579 \DeclareFontShapeChangeRule {scls}{sl} {scls} {}
580 \DeclareFontShapeChangeRule {scls}{sw} {scsw} {sc} % or scls?
581 \DeclareFontShapeChangeRule {scls}{sc} {scls} {}
582 \DeclareFontShapeChangeRule {scls}{ulc} {sl} {}
583 \DeclareFontShapeChangeRule {scls}{up} {sc} {}

584 %\DeclareFontShapeChangeRule {scsw}{n} {n} {}
585 \DeclareFontShapeChangeRule {scsw}{it} {scit} {scsw}
586 \DeclareFontShapeChangeRule {scsw}{sl} {scls} {}
587 \DeclareFontShapeChangeRule {scsw}{sw} {scsw} {}
588 \DeclareFontShapeChangeRule {scsw}{sc} {scsw} {}
589 \DeclareFontShapeChangeRule {scsw}{ulc} {sw} {}
590 \DeclareFontShapeChangeRule {scsw}{up} {sc} {}

591 %\DeclareFontShapeChangeRule {sw}{n} {n} {}
592 %\DeclareFontShapeChangeRule {sw}{it} {it} {}
593 %\DeclareFontShapeChangeRule {sw}{sl} {sl} {}
594 %\DeclareFontShapeChangeRule {sw}{sw} {sw} {}
595 \DeclareFontShapeChangeRule {sw}{sc} {scsw} {}
596 \DeclareFontShapeChangeRule {sw}{ulc} {sw} {}
597 \DeclareFontShapeChangeRule {sw}{up} {n} {}

```

Supporting rollback ...

```

598 //2ekernel | latexrelease)
599 <latexrelease>\EndIncludeInRelease
600 <latexrelease>\IncludeInRelease{0000/00/00}%
601 <latexrelease> {\DeclareFontShapeChangeRule}{Font shape change rules}%
602 <latexrelease>
603 <latexrelease>\let\DeclareFontShapeChangeRule\@undefined
604 <latexrelease>\let\ulcshape\@undefined
605 <latexrelease>\let\ulcdefault\@undefined
606 <latexrelease>\let\swshape\@undefined

```

```

607 〈\latexrelease〉\let\swdefault\@undefined
608 〈\latexrelease〉\let\sscshape\@undefined
609 〈\latexrelease〉\let\sscdefault\@undefined
610 〈\latexrelease〉
611 〈\latexrelease〉\EndIncludeInRelease

```

## 2.2 Changing to a new shape

```

612 〈*2ekernel | \latexrelease〉
613 〈\latexrelease〉\IncludeInRelease{2021/06/01}%
614 〈\latexrelease〉{\fontshape}{Font shape change}%

```

**\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`.

```

615 \DeclareRobustCommand\fontshape[1]
616 {\expandafter\def\expandafter\delayed@f@adjustment\expandafter
617 {\delayed@f@adjustment\delayed@merge@font@shape{\#1}}}

```

(*End of definition for \fontshape.*)

**\fontshapeforce** The unconditional version:

```

618 \DeclareRobustCommand\fontshapeforce[1]
619 {\expandafter\def\expandafter\delayed@f@adjustment\expandafter
620 {\delayed@f@adjustment\edef\f@shape{\#1}}}

```

(*End of definition for \fontshapeforce.*)

Supporting rollback ...

```

621 〈/2ekernel | \latexrelease〉
622 〈\latexrelease〉\EndIncludeInRelease
623 〈\latexrelease〉\IncludeInRelease{2020/02/02}%
624 〈\latexrelease〉{\fontshape}{Font shape change}%
625 〈\latexrelease〉
626 〈\latexrelease〉\DeclareRobustCommand\fontshape[1]{\merge@font@shape{\#1}}
627 〈\latexrelease〉\DeclareRobustCommand\fontshapeforce[1]{\edef\f@shape{\#1}}
628 〈\latexrelease〉
629 〈\latexrelease〉\EndIncludeInRelease
630 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
631 〈\latexrelease〉{\fontshape}{Font shape change}%
632 〈\latexrelease〉
633 〈\latexrelease〉\DeclareRobustCommand\fontshape [1]{\edef\f@shape{\#1}}
634 〈\latexrelease〉\let\fontshapeforce\@undefined
635 〈\latexrelease〉
636 〈\latexrelease〉\EndIncludeInRelease
637 〈*2ekernel | \latexrelease〉
638 〈\latexrelease〉\IncludeInRelease{2020/02/02}%
639 〈\latexrelease〉{\merge@font@shape}{Font shape change rules}%

```

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

```

640 \def\merge@font@shape#1{%
641 \expandafter\expandafter\expandafter
642 \merge@font@shape@
643 \csname shape@\f@shape @#1\endcsname
644 {#1}%

```

```

645 \cnil
646 }

```

(End of definition for `\merge@font@shape`.)

`\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.

```

647 \def\merge@font@shape@#1#2#3\cnil{%
648 \def\reserved@a{#3}%
649 \ifx\reserved@a\empty
650 \edef\f@shape{#2}%
651 \else

```

`\reserved@a` is used in `\@font@shape@subst@warning` so we have to define it in addition to do the `\ifcsname` test

```

652 \edef\reserved@a{\f@encoding / \f@family / \f@series/#1}%
653 \ifcsname \reserved@a\endcsname
654 \edef\f@shape{#1}%
655 \else
656 \ifcsname \f@encoding / \f@family / \f@series/#2\endcsname
657 \edef\f@shape{#2}%
658 \@font@shape@subst@warning
659 \else
660 \edef\f@shape{#3}%
661 \@font@shape@subst@warning
662 \fi
663 \fi
664 \fi
665 }

```

(End of definition for `\merge@font@shape@`.)

`\merge@font@shape@without@substitution`

`\merge@font@shape@without@substitution@`

`\delayed@merge@font@shape`

```

666 \def\merge@font@shape@without@substitution#1{%
667 \expandafter\expandafter\expandafter
668 \merge@font@shape@without@substitution@
669 \csname shape@\f@shape \#1\endcsname
670 {#1}%
671 \cnil
672 }

```

```

673 \def\merge@font@shape@without@substitution@#1#2#3\cnil{%
674 \def\reserved@a{#3}%
675 \ifx\reserved@a\empty
676 \edef\f@shape{#2}%
677 \else
678 \edef\f@shape{#1}%
679 \fi
680 }

```

`\let\delayed@merge@font@shape\merge@font@shape@without@substitution`

(End of definition for `\merge@font@shape@without@substitution`,  
`\merge@font@shape@without@substitution@`, and `\delayed@merge@font@shape`.)

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

682 \protected\def\normalshape
683 {\not@math@alphabet\normalshape\relax
684 \fontshape\shapedefault\selectfont}%


```

*(End of definition for \normalshape.)*

### 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.

```
685 \def\reinstall@nfss@defs{%
686 \protected\def\upshape
687 {\not@math@alphabet\upshape\relax
688 \fontshape\updefault\selectfont}%
689 \protected\def\slshape
690 {\not@math@alphabet\slshape\relax
691 \fontshape\sldefault\selectfont}%
692 \protected\def\scshape
693 {\not@math@alphabet\scshape\relax
694 \fontshape\scdefault\selectfont}%
695 \protected\def\itshape
696 {\not@math@alphabet\itshape\mathit
697 \fontshape\itdefault\selectfont}%
698 \protected\def\ulcshape
699 {\not@math@alphabet\ulcshape\relax
700 \fontshape{ulc}\selectfont}%
701 \protected\def\swshape
702 {\not@math@alphabet\swshape\relax
703 \fontshape\swdefault\selectfont}%
704 \protected\def\sscshape
705 {\not@math@alphabet\sscshape\relax
706 \fontshape\sscdefault\selectfont}%
707 }
```

*(End of definition for \reinstall@nfss@defs.)*

Supporting rollback ...

```
708 </2ekernel | latexrelease>
709 <latexrelease>\EndIncludeInRelease
710 <latexrelease>\IncludeInRelease{0000/00/00}%
711 <latexrelease> {\merge@font@shape}{Font shape change rules}%
712 <latexrelease>
713 <latexrelease>\DeclareRobustCommand\fontshape [1]{\edef\f@shape{\#1}}
714 <latexrelease>\let\fontshapeforce@\undefined
715 <latexrelease>
716 <latexrelease>\let\merge@font@shape@\undefined
717 <latexrelease>\let\merge@font@shape@@\undefined
718 <latexrelease>
```

```

719 〈\latexrelease〉\let\merge@font@shape@without@substitution@undefined
720 〈\latexrelease〉\let\merge@font@shape@without@substitution@@undefined
721 〈\latexrelease〉\let\delayed@merge@font@shape@undefined
722 〈\latexrelease〉
723 〈\latexrelease〉\let\normalshape@undefined
724 〈\latexrelease〉

```

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

```

725 〈\latexrelease〉
726 〈\latexrelease〉\let\reinstall@nfss@defs\relax
727 〈\latexrelease〉\EndIncludeInRelease

```

This initializes the 2020/02/02 extensions to NFSS after any changes in the preamble.

```

728 〈*2ekernel | \latexrelease〉
729 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
730 〈\latexrelease〉
731 〈\g@addto@macro\@kernel@after@begindocument@before
732 〈\reinstall@nfss@defs\init@series@setup〉
733 〈/2ekernel | \latexrelease〉
734 〈\latexrelease〉\EndIncludeInRelease

```

The initialization was introduced in 2020/02/02 but

```

735 〈\latexrelease〉\IncludeInRelease{2020/02/02}%
736 〈\latexrelease〉
737 〈\latexrelease〉\AtBeginDocument{\reinstall@nfss@defs\init@series@setup}
738 〈\latexrelease〉\EndIncludeInRelease
739 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
740 〈\latexrelease〉
741 〈\latexrelease〉\EndIncludeInRelease
742 〈*2ekernel〉
743 〈/2ekernel〉

```

# File 26

## 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 L<sup>A</sup>T<sub>E</sub>X will stop.

### 2 A driver for this document

The next bit of code contains the documentation driver file for T<sub>E</sub>X, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

When this file is processed directly by L<sup>A</sup>T<sub>E</sub>X this will produce the documentation as well.

```
1 <*driver>
2 \documentclass{ltxdoc}
3
4
5 %\OnlyDescription % comment out for implementation details
6
7 \begin{document}
8 \DocInput{ltfsstrc.dtx}
9 \end{document}
10 </driver>
```

### 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.

```
11 <*package>
12 %\NeedsTeXFormat{LaTeX2e}
13 %\ProvidesPackage{tracefnt}[??/?/? v?.??
14 %
15 </package>
```

The debug module makes use of commands contained in a special package file named `trace.sty`.<sup>33</sup>

```
16 <+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.

```
17 <*2ekernel>
18 \def\tracingfonts{%
19 \@font@warning{Command \noexpand\tracingfonts
20 not provided.\MessageBreak
21 Use the ‘tracefnt’ package.\MessageBreak Command found:}%
22 \count@}
23 </2ekernel>
```

The `\count@` in the line above will remove the number after `\tracingfonts`. Note that this definition will be overwritten by the next line if one of these modules are included.

```
24 <*package,trace,debug>
25 \newcount\tracingfonts
26 \tracingfonts=0
27 </package,trace,debug>
```

(End of definition for `\tracingfonts`.)

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

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

---

<sup>33</sup>This package is not in distribution at the moment (and probably doesn't work any longer). Think of this part of the code as being historical artifacts.

```

33 \def\@font@warning#1{%
34 \GenericInfo{(Font)}{\spaces\spaces\space\space}%
35 {LaTeX Font Warning: #1}}%
36 }

37 \DeclareOption{warningshow}{%
38 \def\@font@info#1{%
39 \GenericInfo{(Font)}{\spaces\spaces\space\space}%
40 {LaTeX Font Info: \space\space\space#1}}%
41 \def\@font@warning#1{%
42 \GenericWarning{(Font)}{\spaces\space\space\space}%
43 {LaTeX Font Warning: #1}}%
44 }

45 \DeclareOption{infoshow}{%
46 \def\@font@info#1{%
47 \GenericWarning{(Font)}{\spaces\space\space\space}%
48 {LaTeX Font Info: \space\space\space#1}}%
49 \def\@font@warning#1{%
50 \GenericWarning{(Font)}{\spaces\space\space\space}%
51 {LaTeX Font Warning: #1}}%
52 }

53 \DeclareOption{loading}{%
54 \tracingfonts\tw@
55 }

56 \DeclareOption{debugshow}{%
57 \ExecuteOptions{infoshow}%
58 \tracingfonts\thr@@
59 }

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

```

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

```

70 \ExecuteOptions{infoshow}
71 \ProcessOptions
72
```

We also need a default definition inside the kernel:

```

73 <*2ekernel>
74 \def\@font@info#1{%
75 \GenericInfo{(Font)}{\spaces\space\space\space}%
76 {LaTeX Font Info: \space\space\space#1}}%
77 \def\@font@warning#1{%
78 \GenericWarning{(Font)}{\spaces\space\space\space}%
79 {LaTeX Font Warning: #1}}%
80
```

## 5 Macros common to `fam.tex` and `tracefnt.sty`

In the first versions of `tracefnt.dtx` some macros of `fam.dtx`<sup>34</sup> 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.

```
81 {*2ekernel | package}
82 \def\extract@font{%
83 \get@external@font
```

Then the external font is loaded and assigned to the font identifier stored inside `\font@name` (for this reason we need `\expandafter`).

```
84 \global\expandafter\font\font@name\external@font\relax
```

When tracing we typeout the internal and external font name.

```
85 {*trace}
86 \ifnum \tracingfonts >@one
87 @font@info{External font '\external@font',
88 loaded as\MessageBreak \font@name}\fi
89
```

Finally we call the corresponding “loading action” macros to finish things. First the font is locally selected to allow the use of `\font` inside the loading action macros.

```
90 \font@name \relax
```

The next two lines execute the “loading actions” for the family and then for the individual font shape.

```
91 \csname \f@encoding+\f@family\endcsname
92 \csname\curr@fontshape\endcsname
93 \relax
94 }
95
```

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 of definition for `\extract@font`.*)

- `\get@external@font` This function tries to find an external font name. It will place the name into the macro `\external@font`. If no font is found it will return the one that was defined via `\DeclareErrorFont`.

```
96 {*2ekernel}
97 \def\get@external@font{%
```

We don’t know the external font name at the beginning.

```
98 \let\external@font\empty
99 \edef\font@info{\expandafter\expandafter\expandafter\string
100 \csname \curr@fontshape \endcsname}%
101 \try@size@range
```

---

<sup>34</sup>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.

```

102 \ifx\external@font\empty
103 \try@size@substitution
104 \ifx\external@font\empty
105 @latex@error{Font \expandafter \string\font@name\space
106 not found}\@eha
107 \error@fontshape
108 \get@external@font
109 \fi\fi
110 }
111
```

(End of definition for `\get@external@font`.)

```

112 <*2ekernel | latexrelease | package>
113 <| latexrelease> \IncludeInRelease{2021/06/01}%
114 <| latexrelease> {\selectfont}{Add hook to \selectfont}%

```

`\selectfont` The macro `\selectfont` is called whenever a font change must take place.

```

115 \DeclareRobustCommand\selectfont
116 {%

```

When `debug` is specified we actually want something like 'undebbug'. 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.

```

117 <+debug> \pushtracing
118 <+debug> \ifnum\tracingfonts<4 \tracingoff
119 <+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`.

```

120 \ifx\f@linespread\baselinestretch \else
121 \set@fontsize\baselinestretch\f@size\f@baselineskip \fi

```

The series and shape updates are only prepared by `\fontseries` and `\fontshape` but not executed until after we are ready to change the font face. This way they happen after a possibly new family is set which is important because they look at the available font faces in that family and alter the selection based on availability. Several calls to `\fontseries` or `\fontshape` are delayed in the order in which they appear, so that by switching them one can work around missing intermediate font faces and avoid substitutions.

We first attempt to do the merge without any substitution. As we might end up with a non-existing font face we may have to restart and therefore save the current values of `\f@series` and `\f@shape` before the merge.

But first we make a quick test to see if there are any delayed actions, because if not it is pointless to make all the assignments and try loading a missing fontshape.

```

122 \ifx\delayed@f@adjustment\empty
123 \else
124 \let\f@shape@saved\f@shape
125 \let\f@series@saved\f@series

```

Then we run the delayed adjustments (which use the `\..@without@substitution` commands):

```
126 \delayed@f@adjustment
```

We then check if the resulting combination is valid but for this we have to make sure that the appropriate .fd is loaded if that hasn't happened so far.

```
127 \maybe@load@fontshape
128 \ifcsname \f@encoding/\f@family/\f@series/\f@shape \endcsname
```

If this macro is defined then we are good and no further action is necessary.

Otherwise the combination is not valid, so we redo the merge but this time with substitutions.

```
129 \else
130 \let\f@shape\f@shape@saved
131 \let\f@series\f@series@saved
132 \let\delayed@merge@font@shape\merge@font@shape
133 \let\delayed@merge@font@series\merge@font@series
134 \delayed@f@adjustment
135 \let\delayed@merge@font@shape\merge@font@shape@without@substitution
136 \let\delayed@merge@font@series\merge@font@series@without@substitution
137 \fi
```

Now the series and shape values are updated and we clear `\delayed@f@adjustment`. This is important because on the next execution of `\selectfont` we should not mistakenly redo the delayed actions if there wasn't any series or shape change.

```
138 \let\delayed@f@adjustment\empty
139 \fi
```

If the series was forced we should now cancel that in case the next series change is done with some low-level setting to `\f@series`.

```
140 \forced@seriesfalse
```

Then we generate the internal name of the font by concatenating *family*, *series*, *shape*, and current *size*, with slashes as delimiters between them. This is much more readable than standard L<sup>A</sup>T<sub>E</sub>X's `\twfbf`, etc. We define `\font@name` globally, as always. The reason for this is explained later on.

```
141 \xdef\font@name{%
142 \csname\curr@fontshape/\f@size\endcsname}%
```

We call the macro `\pickup@font` which will load the font if necessary.

```
143 \pickup@font
```

Then we select the font.

```
144 \font@name
```

After switching fonts we run a hook, so that packages can make last minute alterations based on the new font (originally provided in `everysel` but using a different interface).

```
145 \UseHook{selectfont}%
```

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.

```
146 \size@update
```

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

```
147 \enc@update
```

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

```
148 <+debug> \poptracing
149 }
```

*(End of definition for `\selectfont`.)*

**selectfont** Declare the hook used in `selecfont` in the kernel, but not inside the `tracefnt` package.

```
150 <-trace> \NewHook{selectfont}
```

*(End of definition for `selectfont`.)*

If `\tracingfonts` is greater than 2 we also show the font switch inside `\selectfont`. We do this by adding this code to the hook in the `tracefnt` package: macro might redefine `\font@name`.

```
151 <*trace>
152 \AddToHook{selectfont}
153 {\ifnum \tracingfonts>\tw@
154 \font@info{Switching to \font@name}\fi}
155
```

```
156 </2ekernel | latexrelease | package>
157 \latexrelease\EndIncludeInRelease
```

With `\selectfont` having different definitions in different kernels we also have to provide them in the `tracefnt` package to support rollback. In packages that works a bit differently and therefore we have to provide an empty block there.

```
158 <package>\IncludeInRelease{2021/06/01}%
159 <package> {\selectfont}{Add hook to \selectfont}%
160 <package>\EndIncludeInRelease

161 <\latexrelease | package>\IncludeInRelease{0000/00/00}%
162 <\latexrelease | package> {\selectfont}{Add hook to \selectfont}%
163 <\latexrelease | package>
164 <\latexrelease | package>\DeclareRobustCommand\selectfont
165 <\latexrelease | package> {%
166 <\latexrelease | package> \ifx\f@linespread\baselinestretch \else
167 <\latexrelease | package> \set@fontsize\baselinestretch\f@size\f@baselineskip \fi
168 <\latexrelease | package> \xdef\font@name{%
169 \csname curr@fontshape/\f@size\endcsname}%
170 <\latexrelease | package> \pickup@font
171 <\latexrelease | package> \font@name
172 <\latexrelease | package> \size@update
173 <\latexrelease | package> \enc@update
174 <\latexrelease | package> }
175 <\latexrelease | package>
176 <\latexrelease | package>\EndIncludeInRelease
```

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

```
177 <*2ekernel | package>
178 \def\set@fontsize#1#2#3{%
179 \Qdefaultunits\Qtempdimb#2pt\relax\Qnil
```

```

180 \edef\f@size{\strip@pt\@tempdimb}%
181 \@defaultunits\@tempskipa#3pt\relax\@nnil
182 \edef\f@baselineskip{\the\@tempskipa}%
183 \edef\f@linespread{\#1}%

```

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

```
184 \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:

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

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

```

186 \baselineskip\f@baselineskip\relax
187 \baselineskip\f@linespread\baselineskip
188 \normalbaselineskip\baselineskip

```

then to set up a new `\strutbox`

```

189 \setbox\strutbox\hbox{%
190 \vrule\@height.7\baselineskip
191 \@depth.3\baselineskip
192 \@width\z@}%

```

We end with a bit of tracing information.

```

193 <*>
194 \ifnum \tracingfonts>\tw@
195 \ifx\f@linespread\empty
196 \let\reserved@a\empty
197 \else
198 \def\reserved@a{\f@linespread x}%
199 \fi
200 \font@info{Changing size to \f@size/\reserved@a
201 \f@baselineskip}%
202 \aftergroup\type@restoreinfo \fi
203 </>

```

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

```

204 \let\size@update\relax}%
205 }

```

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

*(End of definition for `\set@fontsize`.)*

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

```
206 \let\size@update\relax
```

*(End of definition for `\size@update`.)*

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

```
207 {*trace}
208 \def\type@restoreinfo{%
209 \ifx\f@linespread\empty
210 \let\reserved@a\empty
211 \else
212 \def\reserved@a{\f@linespread x}%
213 \fi
214 \font@info{Restoring size to
215 \f@size/\reserved@a\f@baselineskip}%
216 }
```

(End of definition for `\type@restoreinfo`.)

`\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.

```
218 \expandafter\ifx\csname S@\f@size\endcsname\relax
219 \calculate@math@sizes
220 \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.

```
221 \csname S@\f@size\endcsname
222 \ifmath@fonts
223 {*trace}
224 \ifnum \tracingfonts>\tw@
225 \font@info{Setting up math fonts for
226 \f@size/\f@baselineskip}\fi
227 }
```

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`.

```
228 \begingroup
229 \escapechar\m@ne
230 \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.

```
231 \globaldefs\@ne
232 \math@fonts
```

```

233 \let \glb@currsize \f@size
234 \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\one` 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.

```

235 \the\every@math@size

```

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

```

236 {*trace}
237 \else
238 \ifnum \tracingfonts>\tw@
239 \font@info{No math setup for
240 \f@size/\f@baselineskip}\fi
241 {/trace}
242 \fi
243 }
244 {/2ekernel | package}

```

(End of definition for `\glb@settings` and `\glb@currsize`.)

`\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).

```

245 {*}2ekernel}
246 \def\baselinestretch{1}

```

(End of definition for `\baselinestretch`.)

`\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.

```

247 \newtoks\every@math@size
248 \every@math@size={}
249 {/2ekernel}

```

(End of definition for `\every@math@size`.)

## 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:

- At any point in the document the global variable `\glb@currsize` contains the point size for which the math fonts currently are set up.

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

The above algorithm has the following features:

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

### 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.

```

250 {*2ekernel | package}
251 \def\check@mathfonts{%
252 \ifx \gbl@currsize \f@size
253 {*trace}
254 \ifnum \tracingfonts>\thr@@
255 \o@font@info{*** MATH: no change \f@size\space
256 curr/global (\curr@math@size/\gbl@currsize)}\fi
257 {*}trace}
258 \else
259 {*trace}
260 \ifnum \tracingfonts>\thr@@
261 \o@font@info{*** MATH: setting up \f@size\space
262 curr/global (\curr@math@size/\gbl@currsize)}\fi
263 {*}trace}

```

```

264 \glb@settings
265 \init@restore@glb@settings
266 \fi
267 \let\curr@math@size\f@size
268 \def\init@restore@glb@settings{\aftergroup\restglb@settings}%
269 }

(End of definition for \check@mathfonts.)
```

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

```

270 <-trace> \let\init@restore@glb@settings\relax
271 <*trace>
272 \def\init@restore@glb@settings{%
273 \ifnum \tracingfonts>\thr@@
274 \o@font@info{*** MATH: no resetting (not in
275 nested math)}\fi
276 }
277 </trace>
```

(End of definition for `\init@restore@glb@settings`.)

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

```

278 \def\restglb@settings{%
279 <*trace>
280 \ifnum \tracingfonts>\thr@@
281 \o@font@info{*** MATH: restoring}\fi
282 </trace>
283 \begingroup
284 \let\f@size\curr@math@size
285 \ifx\glb@currsize\f@size
286 <*trace>
287 \ifnum \tracingfonts>\thr@@
288 \o@font@info{*** MATH: ... already okay (\f@size)}\fi
289 </trace>
290 \else
291 <*trace>
292 \ifnum \tracingfonts>\thr@@
293 \o@font@info{*** MATH: ... to \f@size}\fi
294 </trace>
295 \glb@settings
296 \fi
297 \endgroup
298 }
```

(End of definition for `\restglb@settings`.)

### 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.

```
299 \def\use@mathgroup#1#2{\relax\ifmmode
```

```

300 {*trace}
301 \ifnum \tracingfonts>\tw@
302 \count@#2\relax
303 \@font@info{Using \noexpand\mathgroup
304 (\the\count@) #2}\fi
305
```

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.

```

306 \math@bgroup
307 \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
308 #1\fi
309 \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.

```
310 \expandafter\math@egroup\fi}%

```

The surrounding macros equal `\begingroup` and `\endgroup`. But using internal names makes it possible to overwrite their meaning in certain cases. This is for example used in *AMS-T<sub>E</sub>X* macros for placing accents.

*(End of definition for `\use@mathgroup`.)*

`\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`.

```

311 {*trace}
312 \ifx\math@bgroup\bgroup
313 \def\math@egroup#1{\#1\egroup
314 \ifnum \tracingfonts>\tw@
315 \@font@info{Restoring \noexpand\mathgroup
316 (\ifnum\mathgroup=\m@ne default\else \the\mathgroup \fi)%
317 }\fi}
318 \fi
319
```

*(End of definition for `\math@egroup`.)*

`\getanddefine@fonts` `\getanddefine@fonts` has two arguments: the `\math group number` and the *family/series/shape* name as a control sequence.

```
320 \def\getanddefine@fonts#1#2{%

```

First we turn of tracing when `\tracingfonts` is less than 4.

```
321 <+debug> \pushtracing
322 <+debug> \ifnum\tracingfonts<4 \tracingoff
323 <+debug> \else \tracingon\getanddefine@fonts \fi

324 <*trace>
325 \ifnum \tracingfonts>\tw@
326 \count@#1\relax
327 \@font@info{\noexpand\mathgroup (\the\count@) #1 :=\MessageBreak
328 \string#2 \tf@size/\sf@size/\ssf@size}\fi
329
```

We append the current `\tf@size` to #2 to obtain the font name.<sup>35</sup> Again, `font@name` is defined globally, for the reasons explained in the description of `\wrong@fontshape`.

```
330 \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`.

```
331 \pickup@font \let\textfont@name\font@name

```

Same game for `\scriptfont` and `\scriptscriptfont`:

```
332 \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
333 \pickup@font \let\scriptfont@name\font@name
334 \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
335 \pickup@font

```

Then we append the new `\textfont...` assignments to the `\math@fonts`.

```
336 \edef\math@fonts{\math@fonts
337 \textfont#1\textfont@name
338 \scriptfont#1\scriptfont@name
339 \scriptscriptfont#1\font@name}%

```

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

```
340 <+debug> \poptracing
341 }
342
```

(*End of definition for `\getanddefine@fonts`.*)

## 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.

```
343 <*2ekernel>
344 \def\ifnot@nil#1{\def\reserved@a{#1}%
345 \ifx\reserved@a\@nil \expandafter\@gobble
346 \else \expandafter\@firstofone\fi}

```

(*End of definition for `\ifnot@nil`.*)

---

<sup>35</sup>One might ask why this expansion does not generate a macro name that starts with an additional `\` character. The solution is that `\escapechar` is set to `-1` before `\getanddefine@fonts` is called.

\remove@to@nnil Three other auxiliary macros will be needed in the following: \remove@to@nnil gobbles up everything up to, and including, the next \nnil token, and \remove@angles and \remove@star do the same for the character > and \*, respectively, instead of \nnil.

```
347 \def\remove@to@nnil#1\@nil{%
348 \def\remove@angles#1>{\set@simple@size@args}%
349 \def\remove@star#1*{#1}
```

(End of definition for \remove@to@nnil, \remove@angles, and \remove@star.)

\extract@sizefn This macro takes a size specification and parses it into size function and the optional and mandatory arguments.

```
350 \def\extract@sizefn#1*#2\@nil{%
351 \if>#2>\set@size@funct@args#1\@nil
352 \let\sizefn@info\@empty
353 \else\expandafter\set@size@funct@args\remove@star#2\@nil
354 \def\sizefn@info{#1}\fi
355 }
```

(End of definition for \extract@sizefn.)

\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\@nil{%
 \set@simple@size@args#3<#4\@nil
 \execute@size@function{#2}}
```

so that when it gets called via

```
\extract@fontinfo<10*>cmss10<12*>cmss12<17*>cmss17\@nil
```

#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\@nil
\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\@nil{%
 \ifnot@nil{#3}%
 {\set@simple@size@args#3<#4\@nil
 \execute@size@function{#2}%
 }%
}
```

How does this work? We call `\extract@fontinfo` via

```
\expandafter\extract@fontinfo\font@info<12*>\@nil<\@nil
```

i.e. by appending `<12*>\@nil<\@nil`. 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<\@nil`. 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`.

```
356 % % this could be replaced by \try@size@range making the subst slower!
357 \def\try@simple@size{%
```

`\reserved@a` is made an abbreviation for the head of the definition of the macro `\extract@fontinfo`.

```
358 \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.

```
359 \expandafter\reserved@a\expandafter<\f@size>##2<##3\@nil{%
360 \ifnot@nil{##2}%
361 {\set@simple@size@args##2<##3\@nil
362 \execute@size@function\sizefn@info
363 }%
}
```

Now we call `\extract@fontinfo`. Note the `<\@nil` tokens at the end.

```
364 \expandafter\expandafter
365 \expandafter\extract@fontinfo\expandafter\font@info
366 \expandafter<\f@size>\@nil<\@nil
367 }
```

(End of definition for \try@simple@size.)

- \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,

368 \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.

```
369 \if<#1<%
370 \expandafter\remove@angles
371 \else
372 \extract@sizefn#1*\@nil
373 \expandafter\remove@to@nnil
374 \fi}
```

(End of definition for \set@simple@size@args.)

Now, we are through with the case of a simple size, except for calling the size function. This will be handled later, as it is the same mechanism for all types of size specification. We will now proceed to macros for extraction of size range specification.

- \extract@rangefontinfo \extract@rangefontinfo goes through a font shape definition in the input until it recognizes the tokens <\@nil->. It looks for font ranges with font size functions. Its 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.

```
375 \def\extract@rangefontinfo#1<#2>{%
376 \is@range#2->\@nil#2}
```

(End of definition for \extract@rangefontinfo.)

- \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 (that 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.

```
377 \def\is@range#1-#2\@nil{%
378 \if>#2\expandafter\check@single\else
379 \expandafter\check@range\fi}
```

(End of definition for \is@range.)

- \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.

```
380 \def\check@range#1-#2>#3<#4\@nnil{%
381 \ifnot@nil{#3}{%
```

If #3 wasn't `\cnil` 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.

```
382 \def\reserved@f{\extract@rangefontinfo<#4\cnil}%
```

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.

```
383 \upper@bound0#2\p@
384 \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.

```
385 \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.

```
386 \lower@bound0#1\p@
387 \ifdim \f@size \p@<\lower@bound
388 \else
```

If both tests are passed we can try executing the size function.

```
389 \set@simple@size@args#3<#4\cnil
390 \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.

```
391 \ifx\external@font\empty
392 \else
393 \let\reserved@f\empty
394 \fi
395 \fi
396 \fi
397 \reserved@f}}
```

*(End of definition for `\check@range`.)*

`\lower@bound` We use two dimen registers `\lower@bound` and `\upper@bound` to store the lower and upper endpoints of the range we found.

```
398 \newdimen\lower@bound
399 \newdimen\upper@bound
```

*(End of definition for `\lower@bound` and `\upper@bound`.)*

`\check@singl` `\check@singl` 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).

```
400 \def\check@singl#1>#2<#3\cnil{%
```

We start by assuming that we have to recurse. Note that we have to reinsert an `<` as it was already removed by scanning.

```
401 \def\reserved@f{\extract@rangefontinfo<#3\cnil}%
```

Now we check the size against `\f@size`. If it is not equal `\f@size` it is no good and we have to recurse.

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

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

```
403 \set@size@args#2<#3\@nil
404 \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.

```
405 \ifx\external@font\@empty
406 \else
407 \let\reserved@f\@empty
408 \fi
409 \fi
410 \reserved@f}
```

*(End of definition for `\check@single`.)*

`\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`.

```
411 \def\set@size@funct@args{\@ifnextchar[%%
412 \set@size@funct@args@{\set@size@funct@args@[]}}
413 \def\set@size@funct@args@[#1]#2\@nil{%
414 \def\mandatory@arg{#2}%
415 \def\optional@arg{#1}%
416 }/2ekernel}
```

*(End of definition for `\set@size@funct@args` and `\set@size@funct@args@`.)*

`\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 `<...>`.

```
417 /*2ekernel*/
418 \def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
419 \@onlypreamble\DeclareSizeFunction
420 }/2ekernel
```

*(End of definition for `\DeclareSizeFunction`.)*

`\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`).

```
421 /*2ekernel | package)
422 \def\execute@size@function#1{%
423 <*trace>
424 \@ifundefined{s@fct@#1}%
425 {\errmessage{Undefined font size function #1}%
426 \s@fct@}%
427 {\csname s@fct@#1\endcsname}%
428 </trace>
429 {-trace} \csname s@fct@#1\endcsname
430 }
431 }/2ekernel | package)
```

(End of definition for \execute@size@function.)

- \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.

```
432 {*2ekernel}
433 \def\try@size@range{%
434 \expandafter\extract@rangefontinfo\font@info <-*>\@nil<\@nnil
435 }
```

(End of definition for \try@size@range.)

- \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.

```
436 \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.

```
437 \tempdimb \maxdimen
438 \let \best@size \empty
```

Now we loop over the specification

```
439 \expandafter \try@simples \font@info <\number\@M\@nil<\@nnil
440 }
```

(End of definition for \try@size@substitution.)

- \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).

```
441 \def\font@submax{0pt}
442 \def\fontsubfuzz{.4pt}
443 /2ekernel}
444 (+package)\def\fontsubfuzz{0pt}
```

(End of definition for \font@submax and \fontsubfuzz.)

- \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.

```
445 {*2ekernel}
446 \gdef\try@simples#1<#2>{%
447 \tryif@simple#2->\tryif@simple}
```

(End of definition for \try@simples.)

\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.

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

Most common case for \reserved@f first:

```
449 \let \reserved@f \try@simples
450 \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.

```
451 \dimen@ #1\p@
452 \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.

```
453 \ifdim \f@size\p@<\dimen@
454 \tempdimc \dimen@
455 \advance\tempdimc -\f@size\p@
456 \else
457 \tempdimc \f@size\p@
458 \advance\tempdimc -\dimen@
459 \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.

```
460 \ifdim \tempdimc<\tempdimb
461 \tempdimb \tempdimc
462 \def \best@size{#1}%
463 \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.

```
464 \else
```

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

```
465 % %\subst@size %% coded inline
466 % %\def\subst@size{%
467 \ifx \external@font\empty
468 \ifx \best@size\empty
469 \else
470 \ifdim \tempdimb>\font@submax \relax
471 \xdef \font@submax {\the\tempdimb}%
472 \fi
473 \let \f@user@size \f@size
474 \let \f@size \best@size
```

```

475 \ifdim \tempdime>\fontsubfuzz\relax
476 \font@warning{Font\space shape\space
477 '\curr@fontshape'\space in\space size\space
478 <\f@user@size>\space not\space available\MessageBreak
479 size\space <\f@size>\space substituted}%
480 \fi
481 \try@simple@size
482 \do@subst@correction
483 \fi
484 \fi
485 % %}

```

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

```

486 \let \reserved@f \remove@to@nnil
487 \fi
488 \fi

```

If it's a range iterate also.

```
489 \reserved@f}
```

*(End of definition for `\tryis@simple` and `\subst@size`.)*

## 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.

```

490 \DeclareSizeFunction{}{\empty@sfcnt\font@warning}
491 \DeclareSizeFunction{s}{\empty@sfcnt\font@info}
492 \def\empty@sfcnt#1{%
493 \tempdime \f@size\p@
494 \ifx\optional@arg\empty
495 \else
496 \tempdime \optional@arg\tempdime
497 #1{Font\space shape\space '\curr@fontshape'\space
498 will\space be\MessageBreak
499 scaled\space to\space size\space \the\tempdime}%
500 \fi
501 \edef\external@font{\mandatory@arg\space at\the\tempdime}}

```

*(End of definition for `\s@fct@`.)*

- `\s@fct@gen`** This size function generates the external name from the mandatory argument and the requested user size, and thus can be used for external names where the size is encoded in the font name. The optional argument a scale factor. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```

502 \DeclareSizeFunction{gen}{\gen@sfcnt\font@warning}
503 \DeclareSizeFunction{sgen}{\gen@sfcnt\font@info}

```

```

504 \def\gen@sfcnt{%
505 \edef\mandatory@arg{\mandatory@arg\f@size}%
506 \empty@sfcnt}

```

(End of definition for `\s@fct@gen` and `\s@fct@sgen`.)

- `\s@fct@genb` This size function is similar to `gen`, but for fonts where the size is encoded in the font name in centipoins, 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.

```

507 \DeclareSizeFunction{genb}{\genb@sfcnt \@font@warning}
508 \DeclareSizeFunction{sgenb}{\genb@sfcnt \@font@info}
509 \def\genb@sfcnt{%
510 \edef\mandatory@arg{\mandatory@arg\expandafter\genb@x\f@size..\@C}%
511 \empty@sfcnt}

```

(End of definition for `\s@fct@genb` and `\s@fct@sgenb`.)

- `\genb@x` The auxiliary macros `\genb@x` and `\genb@y` are used to convert the `\f@size` into centi-points.

```

512 \def\genb@x#1.#2.#3\@{\two@digits{#1}\genb@y#200\@}
513 \def\genb@y#1#2#3\@{\#1#2}

```

(End of definition for `\genb@x` and `\genb@y`.)

- `\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.

```

514 \DeclareSizeFunction{sub}{\sub@sfcnt \@font@warning}
515 \DeclareSizeFunction{ssub}{\sub@sfcnt \@font@info}
516 \def\sub@sfcnt#1{%
517 \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.

```

518 \begingroup
519 \expandafter\split@name\mandatory@arg/\@nil
520 \try@load@fontshape
521 \endgroup

```

Then we record the current `\f@size` since it may get clobbered.

```

522 \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`.

```

523 \expandafter
524 \ifx\csname\mandatory@arg\endcsname\relax
525 \errmessage{No\space declaration\space for\space
526 shape\space \mandatory@arg}%
527 \error@fontshape
528 \else

```

Otherwise we warn the user about the substitution taking place.

```
529 #1{Font\space shape\space '\curr@fontshape'\space in\space
530 size\space <\f@size>\space not\space available\MessageBreak
531 Font\space shape\space '\mandatory@arg'\space tried\space
532 instead}%
533 \expandafter\split@name\mandatory@arg/\@nil
534 \fi
```

Then we restart the font specification scan by calling `\get@external@font`.

```
535 \edef\f@size{\f@user@size}%
536 \get@external@font
```

Finally `\do@subst@correction` is called to get the font name right.

```
537 \do@subst@correction
538 }
```

*(End of definition for `\s@fct@sub`.)*

`\@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.

```
539 </2ekernel>
540 <*2ekernel | latexrelease>
541 <latexrelease> \IncludeInRelease{2020/02/02}%
542 <latexrelease> {\@font@aliasinfo}{alias size function}%
543 \DeclareSizeFunction{alias}{\sub@sfcnt@\font@aliasinfo}
544 \def@\font@aliasinfo#1{%
545 \@font@info{Font\space shape\space '\curr@fontshape'\space
546 aliased\space to\MessageBreak '\mandatory@arg'}%
547 }
548 </2ekernel | latexrelease>
549 <latexrelease> \EndIncludeInRelease
550 <latexrelease> \IncludeInRelease{0000/00/00}%
551 <latexrelease> {\@font@aliasinfo}{alias size function}%
552 <latexrelease> \let\s@fct@alias@\undefined
553 <latexrelease> \let@\font@aliasinfo@\undefined
554 <latexrelease>
555 <latexrelease> \EndIncludeInRelease
556 <*2ekernel>
```

*(End of definition for `\@font@aliasinfo`.)*

`\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.

```
557 \DeclareSizeFunction{subf}{\subf@sfcnt@\font@warning}
558 \DeclareSizeFunction{ssubf}{\subf@sfcnt@\font@info}
```

```

559 \def\subf@sfcnt#1{%
560 #1{Font\space shape\space '\curr@fontshape'\space in\space
561 size\space \f@size\space not\space available\MessageBreak
562 external\space font\space '\mandatory@arg'\space used}%
563 \empty@sfcnt#1%
564 }

```

*(End of definition for \s@fct@subf.)*

- \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.

```

565 \DeclareSizeFunction{fixed}{\fixed@sfcnt@\font@warning}
566 \DeclareSizeFunction{sfixed}{\fixed@sfcnt@\font@info}
567 \def\fixed@sfcnt#1{%
568 \ifx\optional@arg\empty
569 \let\external@font\mandatory@arg
570 \else
571 \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
572 \fi
573 #1{External\space font\space '\external@font'\space loaded\space
574 for\space size\MessageBreak
575 <\f@size>}%
576 }
577
```

*(End of definition for \s@fct@fixed.)*

## File 27

# ltfsscmp.dtx

This file contains the implementation of commands giving compatibility with the original ‘NFSS1’ release of the Font Selection Scheme.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

Version 1 of NFSS is obsolete now for about 20 years (and was “current” only for a short intermediate time) so with the 2015 release these internal interface commands are removed from the kernel and made available via `latexrelease` package so that backward compatibility remains ensured for very old documents.

```
1 {*latexrelease}
2 \IncludeInRelease{2015/01/01}{\new@fontshape}%
3 {NFSS version1 commands}%
4 \let\new@fontshape@undefined
5 \let\warn@rel@i@undefined
6 \let\scan@fontshape@undefined
7 \let\scan@@fontshape@undefined
8 \let\subst@fontshape@undefined
9 \let\extra@def@undefined
10 \let\default@mextra@undefined
11 \let\preload@sizes@undefined
12 \let\err@rel@i@undefined
13 \let\newmathalphabet@undefined
14 \let\newmathalphabet@undefined
15 \let\newmathalphabet@@@@undefined
16 \let\if@no@font@opt@undefined
17 \let@no@font@optfalse@undefined
18 \let\define@mathalphabet@undefined
19 \let\define@mathgroup@undefined
20 \let\addtoversion@undefined
21 \EndIncludeInRelease
```

In older releases we provide the original definitions.

```
22 \IncludeInRelease{0000/00/00}{\new@fontshape}%
23 {NFSS version1 commands}%
```

`\new@fontshape` The interface is now `\DeclareFontShape`.

```
24 \gdef\new@fontshape#1#2#3#4{%
25 \warn@rel@i\new@fontshape\DeclareFontShape
26 \expandafter\scan@fontshape\@gobble#4<\@nil><<%
27 \DeclareFontShape U{#1}{#2}{#3}\reserved@f}%
28 \onlypreamble\new@fontshape
```

(End of definition for `\new@fontshape`.)

`\warn@rel@i` The warning message used above.

```
29 \gdef\warn@rel@i#1#2{%
30 \font@warning{*** NFSS release 1 command}
```

```

31 \noexpand#1found\MessageBreak
32 *** Update by using release 2 command
33 \string#2.\MessageBreak
34 *** Recovery is probably possible}%
35 }%
36 \onlypreamble\warn@rel@i

```

(End of definition for \warn@rel@i.)

\scan@fontshape This will scan the old font shape definition syntax.

```

37 \gdef\scan@fontshape{%
38 \let\reserved@f\empty
39 \let\reserved@e\empty % holds last info
40 \scan@@fontshape
41 }%
42 \onlypreamble\scan@fontshape

```

(End of definition for \scan@fontshape.)

\scan@@fontshape

```

43 \gdef\scan@@fontshape#1>#2#3<%
44 \ifx\@nil#1%
45 \edef\reserved@f{\reserved@f\reserved@e}%
46 \else
47 \def\reserved@b{#1}% nick names
48 \def\reserved@c{#3}%
49 \in@{ at}{#3}%
50 \ifin@
51 \in@{pt}{#3}%
52 \ifin@ not a proof but a good chance

```

We grab also everything after pt and discard it if people have forgotten to place a percent sign there.

```

53 \def\reserved@a##1 at##2pt##3\@nil{%
54 \def\reserved@b{##2}%
55 \def\reserved@c{##1}%
56 }%
57 \reserved@a#3\@nil
58 \fi
59 \fi
60 \ifnum 0<0#2
61 \edef\reserved@d{\subf*\reserved@c}%
62 \ifcase #2\or
63 \or
64 \else
65 \errmessage{*** What's this? NFSS release 0? ***}%
66 \fi
67 \else
68 \edef\reserved@d{#2\reserved@c}%
69 \fi
70 \ifx\reserved@d\reserved@e
71 \edef\reserved@f{\reserved@f<\reserved@b>}%
72 \else
73 \edef\reserved@f{\reserved@f\reserved@e<\reserved@b>}%add old info
74 \let\reserved@e\reserved@d

```

```

75 \fi
76 \expandafter\scan@@fontshape
77 \fi
78 }%
79 \onlypreamble\scan@@fontshape

```

(End of definition for `\scan@@fontshape`.)

`\subst@fontshape` This is now also handled by the extend syntax of `\DeclareFontShape`.

```

80 \gdef\subst@fontshape#1#2#3#4#5#6{%
81 \warn@rel@i\subst@fontshape\DeclareFontShape
82 \DeclareFontShape{U}{#1}{#2}{#3}{<->sub*#4/#5/#6}{()}%
83 \onlypreamble\subst@fontshape

```

(End of definition for `\subst@fontshape`.)

`\extra@def` This was replaced by `\DeclareFontFamily`.

```

84 \gdef\extra@def#1#2#3{%
85 \warn@rel@i\extra@def\DeclareFontFamily
86 \DeclareFontFamily{U}{#1}{()}%
87 }%
88 \onlypreamble\extra@def

```

(End of definition for `\extra@def`.)

`\default@mextra` The new name is `\DeclareFontEncodingDefaults` but in this case we don't feel comfortable with this either.

```

89 \gdef\default@mextra{%
90 \warn@rel@i\default@mextra\DeclareFontEncodingDefaults

```

We pick up the argument to `\default@mextra` implicitly as the second argument of `\DeclareFontEncodingDefaults`.

```

91 \DeclareFontEncodingDefaults\relax
92 }%
93 \onlypreamble\default@mextra

```

(End of definition for `\default@mextra`.)

`\preload@sizes` The new interface is `\DeclarePreloadSizes`.

```

94 \gdef\preload@sizes{%
95 \warn@rel@i\preload@sizes\DeclarePreloadSizes
96 \DeclarePreloadSizes U%
97 }%
98 \onlypreamble\preload@sizes

```

(End of definition for `\preload@sizes`.)

`\err@rel@i` This macro is used in cases where emulation with NFSS2 features is not really possible.

```

99 \gdef\err@rel@i#1#2{%
100 \o@late@error{*** NFSS release 1 command \noexpand#1found%
101 ^^J*** Recovery not possible. Use \string#2}%
102 {The new release of NFSS doesn't support the
103 \noexpand#1command^^Jany longer.
104 Please upgrade your file to the syntax of NFSS
105 release 2^^Jusing the \noexpand#2command.}%

```

Let's die.

```
106 \batchmode\input.\relax
107 }%
108 \onlypreamble\err@rel@i
```

(End of definition for \err@rel@i.)

\newmathalphabet \newmathalphabet is the old form.

```
109 \gdef\newmathalphabet{%
110 \if@no@font@opt
111 \@latex@error{*** NFSS release 1 command
112 \noexpand\newmathalphabet found%
113 ^^J \space*** Automatic recovery not possible.%}
114 ^^J \space*** TYPE H for Help%
115 }%
116 {Please look at the file usrguide.tex for hints on
117 how to resolve this problem.}%
118 \else
119 \warn@rel@i\newmathalphabet\DeclareMathAlphabet
120 \fi
121 \@ifstar\newmathalphabet@@@
122 \newmathalphabet@@%
123 \gdef\newmathalphabet@@{\DeclareMathAlphabet#1{U}{\{}{\}}{}}%
124 \gdef\newmathalphabet@@@#1#2#3#4{%
125 \DeclareMathAlphabet{#1}{U}{#2}{#3}{#4}}%
126 \onlypreamble\newmathalphabet
127 \onlypreamble\newmathalphabet@@
128 \onlypreamble\newmathalphabet@@@
```

(End of definition for \newmathalphabet , \newmathalphabet@@ , and \newmathalphabet@@@.)

\if@no@font@opt  
\@no@font@optfalse 

```
129 \global\let\if@no@font@opt\iftrue
130 \gdef\@no@font@optfalse{\let\if@no@font@opt\iffalse}%
```

(End of definition for \if@no@font@opt and \@no@font@optfalse.)

\define@mathalphabet This is a case where dying is best.

```
131 \gdef\define@mathalphabet{%
132 \err@rel@i\define@mathalphabet\DeclareMathAlphabet
133 }%
134 \onlypreamble\define@mathalphabet
```

(End of definition for \define@mathalphabet.)

\define@mathgroup And here is another one

```
135 \gdef\define@mathgroup{%
136 \err@rel@i\define@mathgroup\DeclareSymbolFont
137 }%
138 \onlypreamble\define@mathgroup
```

(End of definition for \define@mathgroup.)

```
\addtoversion \addtoversion is the old form.
139 \def\addtoversion#1#2{
140 \warn@rel@i\addtoversion\SetMathAlphabet
141 \SetMathAlphabet#2{#1}{U}}%
142 \onlypreamble\addtoversion

(End of definition for \addtoversion.)
Finishing off this huge \IncludeInRelease argument:
143 \EndIncludeInRelease
144 </latexrelease>
```

## File 28

# ltfssdcl.dtx

This file contains the main implementation of the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

## 1 Interface Commands

\in@ \c@in is a utility macro with two arguments. It determines whether its first argument \ifin@ occurs in its second and sets the switch \ifin@ accordingly. The first argument may not contain braces nor # (more precisely, tokens of category code 1, 2, or 6).

```
1 {*2ekernel}
2 \def\in@#1#2%
3 {%
4 \begingroup
5 \def\in@@##1#1{%
6 \toks@\expandafter{\in@@#2{}{}#1}%
7 \edef\in@{\the\toks@}%
8 }\expandafter\endgroup
9 \ifx\in@{\empty}
10 \in@false
11 \else
12 \in@true
13 \fi
14 }
15 \newif\ifin@
```

(End of definition for \in@ and \ifin@.)

Before the \begin{document} command several *math versions* and *math alphabet identifiers* may be declared. In principle, there should be exactly one family/series/shape combination be declared for each version/alphabet pair. But we want to allow for defaults as well for automagical filling of holes.

While building the tables for math alphabet identifiers and math versions we keep several lists:

- the list of all math versions, \version@list, each entry prefixed by the control sequence \version@elt, i.e. this list has the following form

$$\text{\version@elt}\langle\text{version}_1\rangle\text{\version@elt}\langle\text{version}_2\rangle\dots\text{\version@elt}\langle\text{version}_n\rangle$$

- the list of all math alphabet identifiers. Here every entry has the form:

$$\text{\group@elt}\langle\text{math group number}\rangle\{\{\langle\text{default family}\rangle\}\{\langle\text{default series}\rangle\}\{\langle\text{default shape}\rangle\}\}.$$

- 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>
...
\@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.

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

(*End of definition for \version@list.*)

`\version@elt`

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

(*End of definition for \version@elt.*)

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

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

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

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

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

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

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

```
26 \% \else
27 \% \global\expandafter\newcount\csname c@\expandafter
28 \% \gobble\string#1\endcsname
29 \% \global\csname c@\expandafter
30 \% \gobble\string#1\endcsname\@ne
31 \% \def\version@elt{\noexpand\version@elt\noexpand}%
32 \% \edef\version@list{\version@list\version@elt#1}%
```

Then we prepare to enter the new version into all math alphabet identifier lists. Remember that these lists use `\reserved@c` as delimiter, and that there appears the control sequence `\reserved@e` that must not be expanded. Therefore we take suitable precautions.

```
33 % \def\reserved@c{\noexpand\reserved@c\noexpand}%
34 % \let\reserved@e\relax
```

We will now go through the `\alpha@list` to process every *<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).

```
35 % \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.

```
36 % \edef##1{\expandafter\remove@nil##1%
37 % \reserved@c
38 % #1%
39 % \reserved@e
40 % \noexpand\@nil}}%
```

Finally we call `\alpha@list` which will now execute the macro `\group@elt` once for every defined *<math alphabet identifier>*. And that's all for now.

```
41 % \alpha@list
42 % \fi}
```

(End of definition for `\new@mathversion`.)

`\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`.

```
43 \let\alpha@list\empty
44 \onlypreamble\alpha@list
```

(End of definition for `\alpha@list`.)

`\alpha@elt`

```
45 \let\alpha@elt\relax
46 \onlypreamble\alpha@elt
```

(End of definition for `\alpha@elt`.)

`\newgroup` Start the group (fam) allocation at 0. (Doesn't belong here.)

```
47 \count18=-1
```

(End of definition for `\newgroup`.)

`\stepcounter`

(End of definition for `\stepcounter`.)

\select@group We surround \select@group with braces so that functions using it can be used directly after \_ or ^. However, if we use oldstyle syntax where the math alphabet doesn't have arguments (ie if \math@bgroup is not \bgroup) we need to get rid of the extra group.

```

48 </2ekernel>
49 <latexrelease>\IncludeInRelease{2015/01/01}
50 <latexrelease> {\select@group}{\select@group}%
51 <2ekernel | latexrelease>
52 \def\select@group#1#2#3#4{%
53 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
54 {%
55 \ifmmode
56 \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
57 \begingroup
58 \escapechar\m@ne
59 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
60 \globaldefs\@ne \math@fonts
61 \endgroup
62 \init@restore@version
63 \xdef#1{\noexpand\use@mathgroup\noexpand#2%
64 {\number\csname c@mv@\math@version\endcsname}}%
65 \global\advance\csname c@mv@\math@version\endcsname\@ne
66 \else
67 \let#1\relax
68 \@latex@error{Too many math alphabets used in
69 version \math@version}%
70 \@eha
71 \fi
72 \else \expandafter\@non@alpherr\fi
73 #1{#4}%
74 }%
75 }
76 </2ekernel | latexrelease>
77 <latexrelease>\EndIncludeInRelease
78 <latexrelease>\IncludeInRelease{0000/00/00}
79 <latexrelease> {\select@group}{\select@group}%
80 <latexrelease>\def\select@group#1#2#3#4{%
81 <latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
82 <latexrelease> {%
83 <latexrelease> \ifmmode
84 <latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\sixt@n
85 <latexrelease> \begingroup
86 <latexrelease> \escapechar\m@ne
87 <latexrelease> \getanddefine@fonts
88 <latexrelease> {\csname c@mv@\math@version\endcsname}#3%
89 <latexrelease> \globaldefs\@ne \math@fonts
90 <latexrelease> \endgroup
91 <latexrelease> \init@restore@version
92 <latexrelease> \xdef#1{\noexpand\use@mathgroup\noexpand#2%
93 <latexrelease> {\number\csname c@mv@\math@version\endcsname}}%
94 <latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
95 <latexrelease> \else
96 <latexrelease> \let#1\relax
97 <latexrelease> \@latex@error{Too many math alphabets used in
98 version \math@version}%

```

```

99 \if@eha
100 \fi
101 \else \expandafter\non@alpherr\fi
102 #1{#4}%
103 }%
104 \}
105 \EndIncludeInRelease
106 {*2ekernel}

107 \onlypreamble\restore@mathversion

```

(End of definition for \select@group.)

### \init@restore@version

```

108 \def\init@restore@version{%
109 \global\let\init@restore@version\relax
110 \xdef\restore@mathversion
111 {\expandafter\noexpand\csname mv@\math@version\endcsname
112 \global\csname c@mv@\math@version\endcsname
113 \number\csname c@mv@\math@version\endcsname\relax}%
114 \aftergroup\dorestore@version
115 }
116 \onlypreamble\init@restore@version

```

(End of definition for \init@restore@version.)

### \non@alpherr

```

117 \gdef\non@alpherr#1{\@latex@error{%

```

The command here will have a space at the end of its name, so we make sure not to insert an extra one.

```

118 \string#1allowed only in math mode}\@ehd}

```

(End of definition for \non@alpherr.)

### \dorestore@version

```

119 \def\dorestore@version
120 { \ifmmode
121 \aftergroup\dorestore@version
122 \else
123 \gdef\init@restore@version{%
124 \global\let\init@restore@version\relax
125 \xdef\restore@mathversion
126 {\expandafter\noexpand\csname mv@\math@version\endcsname
127 \global\csname c@mv@\math@version\endcsname
128 \number\csname c@mv@\math@version\endcsname\relax}%
129 \aftergroup\dorestore@version
130 }%
131 \begingroup
132 \let\getanddefine@fonts\@gobbletwo
133 \restore@mathversion
134 \endgroup
135 \fi}%
136 \onlypreamble\dorestore@version

```

(End of definition for \dorestore@version.)

`\c@localmathalphabets` To avoid hitting the “no more math fams available” limit of 16, we keep a defined number of math alphabets flexible/local. If we have to allocate any of those we roll back the allocation after the formula has ended, so the next formula can use other alphabets in the slot(s). This makes the processing a bit slower if you are working at the limit, but that is better than dying with “out of memory”.

```

137 </2ekernel>
138 {latexrelease}\IncludeInRelease{2021/11/15}
139 {latexrelease} {\document@select@group}{\document@select@group}%
140 {*2ekernel | latexrelease}

```

We don’t really undo the declaration on rollback (as that would be hard to maintain), so rolling forward needs to check if the declaration was already made.

```
141 \ifx\c@localmathalphabets\undefined
```

There is no need to have this counter as part of the include checkpoints, given that it makes little sense to alter its settings mid document. All we want is the ability to change it using the `\setcounter` interface.

By default we keep two math fams flexible.

```

142 \newcount\c@localmathalphabets
143 \setcounter{localmathalphabets}{2}
144 \fi

```

(*End of definition for \c@localmathalphabets.*)

`\document@select@group` The `\document@select@group` command is the version of `\select@group` (inside math versions) that is used in the document body to set up math alphabets (if used).

```

145 \def\document@select@group#1#2#3#4{%
146 \ifx\math@bgroup\math@bgroup\else\relax\expandafter\@firstofone\fi
147 {%
148 \ifmmode

```

We first check if there is still room for allocating another mathgroup. If there is, we check if it can be globally allocated or if we have reached the limit which is given by `\e@mathgroup@top` with `\c@localmathalphabets` subtracted.

```

149 \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
150 \ifnum \numexpr\mathgroup@top-\c@localmathalphabets
151 >\csname c@mv@\math@version\endcsname
152 \else

```

If we are past this point we freeze the current state of the math version so that we can return to it after the formula has ended. Of course, that should be done only once, so we check if `\mv@<version>@frozen` already exists.

```
153 \ifcsname mv@\math@version @frozen\endcsname \else
```

We have to pass the current value of `\math@version` not the macro itself, because some of the processing is delayed to a point where the value may have changed again—not doing this caused a puzzling error in one setup.

```

154 \expandafter\freeze@\math@version\expandafter{\math@version}%
155 \fi
156 \fi
157 \begingroup
158 \escapechar\m@ne
159 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
160 \globaldefs\one \math@fonts

```

```

161 \endgroup
162 \expandafter\extract@alph@from@version
163 \csname mv@\math@version\expandafter\endcsname
164 \expandafter{\number\csname
165 c@mv@\math@version\endcsname}%
166 #1%
167 \global\advance\csname c@mv@\math@version\endcsname\@ne
168 \else
169 \let#1\relax
170 \@latex@error{Too many math alphabets used in
171 version \math@version}%
172 \@eha
173 \fi

```

Extra \expandafter to remove the \expandafter added below

```
174 \else \expandafter\expandafter\expandafter\non@alpherr\fi
```

We surround \select@group with braces so that functions using it can be used directly after \_ or ^.

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.

```

175 \expandafter#1\ifx\math@bgroup\bgroup{\#4}\else#4\fi
176 }%
177 }
178 </2ekernel | latexrelease>
179 <| latexrelease>\EndIncludeInRelease
180 <| latexrelease>\IncludeInRelease{2020/10/01}
181 <| latexrelease> {\document@select@group}{\document@select@group}%
182 <| latexrelease>
183 <| latexrelease>\def\document@select@group#1#2#3#4{%
184 <| latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
185 <| latexrelease> {%
186 <| latexrelease> \ifmmode
187 <| latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
188 <| latexrelease> \begin{group}
189 <| latexrelease> \escapechar\m@ne
190 <| latexrelease> \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
191 <| latexrelease> \globaldefs\@ne \math@fonts
192 <| latexrelease> \end{group}
193 <| latexrelease> \expandafter\extract@alph@from@version
194 <| latexrelease> \csname mv@\math@version\expandafter\endcsname
195 <| latexrelease> \expandafter{\number\csname
196 <| latexrelease> c@mv@\math@version\endcsname}%
197 <| latexrelease> #1%
198 <| latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
199 <| latexrelease> \else
200 <| latexrelease> \let#1\relax
201 <| latexrelease> \@latex@error{Too many math alphabets used
202 <| latexrelease> in version \math@version}%
203 <| latexrelease> \@eha
204 <| latexrelease> \fi
205 <| latexrelease> \else \expandafter\expandafter\expandafter\non@alpherr\fi

```

```

206 〈\latexrelease〉 \expandafter#1\ifx\math@bgroup\bgroup{#4}\else#4\fi
207 〈\latexrelease〉 }%
208 〈\latexrelease〉
209 〈\latexrelease〉\EndIncludeInRelease
210 〈\latexrelease〉\IncludeInRelease{2015/01/01}
211 〈\latexrelease〉 {\document@select@group}{\document@select@group}%
212 〈\latexrelease〉
213 〈\latexrelease〉\def\document@select@group#1#2#3#4{%
214 〈\latexrelease〉 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
215 〈\latexrelease〉 {%
216 〈\latexrelease〉 \ifmmode
217 〈\latexrelease〉 \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
218 〈\latexrelease〉 \begin{group}
219 〈\latexrelease〉 \escapechar\m@ne
220 〈\latexrelease〉 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
221 〈\latexrelease〉 \globaldefs\@ne \math@fonts
222 〈\latexrelease〉 \endgroup
223 〈\latexrelease〉 \expandafter\extract@alph@from@version
224 〈\latexrelease〉 \csname mv@\math@version\expandafter\endcsname
225 〈\latexrelease〉 \expandafter{\number\csname
226 〈\latexrelease〉 c@mv@\math@version\endcsname}%
227 〈\latexrelease〉 #1%
228 〈\latexrelease〉 \global\advance\csname c@mv@\math@version\endcsname\@ne
229 〈\latexrelease〉 \else
230 〈\latexrelease〉 \let#1\relax
231 〈\latexrelease〉 \@latex@error{Too many math alphabets used
232 〈\latexrelease〉 in version \math@version}%
233 〈\latexrelease〉 \relax
234 〈\latexrelease〉 \fi
235 〈\latexrelease〉 \else \expandafter\non@alpherr\fi
236 〈\latexrelease〉 #1{#4}%
237 〈\latexrelease〉 }%
238 〈\latexrelease〉
239 〈\latexrelease〉\EndIncludeInRelease
240 〈\latexrelease〉
241 〈\latexrelease〉\IncludeInRelease{0000/00/00}
242 〈\latexrelease〉 {\document@select@group}{\document@select@group}%
243 〈\latexrelease〉
244 〈\latexrelease〉\def\document@select@group#1#2#3#4{%
245 〈\latexrelease〉 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
246 〈\latexrelease〉 {%
247 〈\latexrelease〉 \ifmmode
248 〈\latexrelease〉 \ifnum\csname c@mv@\math@version\endcsname<\sixt@n
249 〈\latexrelease〉 \begin{group}
250 〈\latexrelease〉 \escapechar\m@ne
251 〈\latexrelease〉 \getanddefine@fonts
252 〈\latexrelease〉 {\csname c@mv@\math@version\endcsname}#3%
253 〈\latexrelease〉 \globaldefs\@ne \math@fonts
254 〈\latexrelease〉 \endgroup
255 〈\latexrelease〉 \expandafter\extract@alph@from@version
256 〈\latexrelease〉 \csname mv@\math@version\expandafter\endcsname
257 〈\latexrelease〉 \expandafter{\number\csname
258 〈\latexrelease〉 c@mv@\math@version\endcsname}%
259 〈\latexrelease〉 #1%

```

```

260 <|latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
261 <|latexrelease> \else
262 <|latexrelease> \let#1\relax
263 <|latexrelease> \@latex@error{Too many math alphabets used
264 <|latexrelease> in version \math@version}%
265 <|latexrelease> \@eha
266 <|latexrelease> \fi
267 <|latexrelease> \else \expandafter\non@alpherr\fi
268 <|latexrelease> #1{#4}%
269 <|latexrelease> }%
270 <|latexrelease>}
271 <|latexrelease>\EndIncludeInRelease
272 <|2ekernel>

```

(End of definition for `\document@select@group.`)

`\freeze@math@version` This command stores the current state of the math version and sets things up to return to it after each formula from now on. We use L3 programming layer code to set it up.

```

273 <|2ekernel>
274 <|2ekernel | latexrelease>
275 <|latexrelease>\IncludeInRelease{2022/11/01}%
276 <|latexrelease> \freeze@math@version}{freeze math version}%
277 \ExplSyntaxOn
278 \cs_new_protected:Npn\freeze@math@version #1 {

```

Save the current `\mv@<version>` code and the number of allocated mathgroups inside.

```

279 @font@info{Freeze~ math~ alphabet~ allocation~ in~ version-
280 #1.\MessageBreak
281 Allocated~math~groups:~\int_use:c{ c@mv@ #1 }~
282 (local:~ \int_use:N\c@localmathalphabets) }
283 \cs_gset_eq:cc { mv@#1@frozen }{ mv@#1 }
284 \tl_gset:cx { g__nfss_frozen_mv_ #1 _tl }{ \int_use:c { c@mv@#1 } }

```

Here is the definition of `\mv@<version>@reset`. If there has been no new math alphabet allocation, doing a reset would just cause a lot of unnecessary processing, so we do a quick check upfront for this.

```

285 \cs_gset:cpn{mv@#1@reset}
286 {
287 \int_compare:nNnTF { \int_use:c{c@mv@#1} } >
288 { \tl_use:c{g__nfss_frozen_mv_ #1 _tl} }
289 {
290 @font@info{Undo~ math~ alphabet~ allocation~ in~ version~ #1}

```

If the undo is necessary, we restore the `\mv@<version>` code.

```

291 \cs_gset_eq:cc { mv@#1 }{ mv@#1@frozen }
292 \int_gset:cn { c@mv@#1 }{ \tl_use:c {g__nfss_frozen_mv_ #1 _tl} }

```

But we also should undo changes to the math alphabet definitions. We therefore run this code with a modified definition for `\getanddefine@fonts` because there is no need to do anything to the symbol fonts that are permanently allocated.

```

293 \group_begin:
294 \cs_set_eq:NN \getanddefine@fonts \use_none:nn
295 \use:c {mv@#1}
296 \group_end:
297 }
298 {

```

If there was no change, we report that in the log (but this branch could go completely).

```
299 \@font@info{No~ math~ alphabet~ change~ to~ frozen~ version~ #1}
300 }
```

If this is executed after a math display, we may have to arrange for ignoring spaces, because they are now hidden if the tokens from above intervene. This is signaled by the 2e switch `@ignore` which is set in `\frozen@everymath` and `\frozen@everydisplay`.

This is all 2e code so we use that syntax.

```
301 \if@ignore \ignorespaces \fi
302 }
303 }
304 \ExplSyntaxOff
305 </2ekernel | latexrelease>
306 <latexrelease>\EndIncludeInRelease
307 <latexrelease>\IncludeInRelease{2021/11/15}
308 <latexrelease> {\freeze@math@version}{freeze math version}%
309 <latexrelease>
310 <latexrelease>\ExplSyntaxOn
311 <latexrelease>\cs_set_protected:Npn\freeze@math@version #1 {
312 <latexrelease> \@font@info{Freeze~ math~ alphabet~ allocation~ in~ version~
313 <latexrelease> #1.\MessageBreak
314 <latexrelease> Allocated~math~groups:~\int_use:c{ c@mv@ #1 }~%
315 <latexrelease> (local:~ \int_use:N\c@localmathalphabets) }
316 <latexrelease> \cs_gset_eq:cc { mv@#1@frozen }{ mv@#1 }
317 <latexrelease> \tl_gset:cx { g__nfss_frozen_mv_ #1 _tl }{ \int_use:c { c@mv@#1 } }
318 <latexrelease> \group_insert_after:N __nfss_init_mv_freeze:N
319 <latexrelease> \exp_after:wN \group_insert_after:N \cs:w mv@#1@reset \cs_end:
320 <latexrelease> \tl_gput_right:No \check@mathfonts
321 <latexrelease> {
322 <latexrelease> \exp_after:wN \group_insert_after:N \cs:w mv@#1@reset \cs_end:
323 <latexrelease> }
324 <latexrelease> \cs_gset:cpn{mv@#1@reset}
325 <latexrelease> {
326 <latexrelease> \int_compare:nNnTF { \int_use:c{c@mv@#1} } >
327 <latexrelease> { \tl_use:c{g__nfss_frozen_mv_ #1 _tl} }
328 <latexrelease> {
329 <latexrelease> \@font@info{Undo~ math~ alphabet~ allocation~ in~ version~ #1}
330 <latexrelease> \cs_gset_eq:cc { mv@#1 }{ mv@#1@frozen }
331 <latexrelease> \int_gset:cn { c@mv@#1 }{ \tl_use:c { g__nfss_frozen_mv_ #1 _tl} }
332 <latexrelease> \group_begin:
333 <latexrelease> \cs_set_eq:NN \getanddefine@fonts \use_none:nn
334 <latexrelease> \use:c { mv@#1 }
335 <latexrelease> \group_end:
336 <latexrelease> }
337 <latexrelease> {
338 <latexrelease> \@font@info{No~ math~ alphabet~ change~ to~ frozen~ version~ #1}
339 <latexrelease> }
340 <latexrelease> \if@ignore \ignorespaces \fi
341 <latexrelease> }
342 <latexrelease>
343 <latexrelease>\cs_set_protected:Npn __nfss_init_mv_freeze:N #1 {%
344 <latexrelease> \mode_if_math:T { \group_insert_after:N __nfss_init_mv_freeze:N
345 <latexrelease> \group_insert_after:N } #1
346 <latexrelease>}
```

```

347 ⟨latexrelease⟩\ExplSyntaxOff
348 ⟨latexrelease⟩
349 ⟨latexrelease⟩\EndIncludeInRelease
350 ⟨*2ekernel⟩

```

(End of definition for `\freeze@math@version`.)

### `\process@table`

```

351 \def\process@table{%
352 \def\cdp@elt##1##2##3##4{%
353 \o@font@info{Checking defaults for
354 ##1##2##3##4}%
355 \expandafter
356 \ifx\csname##1##2##3##4\endcsname\relax
357 \begingroup
358 \def\f@encoding{##1}\def\f@family{##2}%
359 \try@load@fontshape
360 \endgroup
361 \fi
362 \expandafter
363 \ifx\csname##1##2##3##4\endcsname\relax
364 \@latex@error{This NFSS system isn't set up properly}%
365 {For encoding scheme ##1 the defaults
366 ##2##3##4 do not form a valid font shape}%
367 \else
368 \o@font@info{... okay}%
369 \fi}%
370 \cdp@list

```

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

```

371 \begingroup
372 \escapechar\m@ne
373 \error@fontshape
374 \expandafter\ifx\csname \curr@fontshape\endcsname\relax
375 \begingroup
376 \try@load@fontshape
377 \endgroup
378 \fi
379 \expandafter\ifx\csname \curr@fontshape\endcsname\relax
380 \@latex@error{This NFSS system isn't set up properly}%
381 {The system maintainer forgot to specify a suitable
382 substitution
383 font shape using the \noexpand\DeclareErrorFont
384 command}%
385 \fi
386 \endgroup

```

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

```

387 \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`.

```

388 \fontencoding\encodingdefault
389 \edef\f@family{\familydefault}%
390 \edef\f@series{\seriesdefault}%
391 \edef\f@shape{\shapedefault}%

```

Drop stuff not longer needed. We need to add many more!!!!!!

```

392 \everyjob{}%
393 }
394 \onlypreamble\process@table

```

(End of definition for `\process@table`.)

```

395 \%onlypreamble\set@mathradical

```

### `\DeclareMathVersion`

```

396 </2ekernel>
397 <*2ekernel | latexrelease>
398 <latexrelease>\IncludeInRelease{2022/11/01}%
399 <latexrelease> {\DeclareMathVersion}{local alphabets}%
400 \def\DeclareMathVersion#1{%

```

When declaring a new math version we need to instantiate an L3 variable that is used when we freeze the version, because too many alphabets got allocated. If we don't do this, L3 programming layer complains if it is run in checking mode.

```
401 \namedef{g_nfss_frozen_mv_#1_t1}{}%
```

We also extend `\check@mathfonts` to call a version reset (once frozen) after a formula has finished.

```

402 \expandafter\ifx\csname mv@#1\endcsname \relax
403 \expandafter \g@addto@macro \expandafter \check@mathfonts
404 \expandafter {\expandafter \aftergroup \csname mv@#1\reset\endcsname}%

```

Initially this macro does nothing. It is, however, important that it doesn't stop any `\ignorespaces`, so we make it expandable and not `\relax`.

```

405 \namedef{mv@#1\reset}{}%
406 \fi
407 \expandafter\new@mathversion\csname mv@#1\endcsname}
408 \onlypreamble\DeclareMathVersion
409 </2ekernel | latexrelease>
410 <latexrelease>\EndIncludeInRelease

```

```

411 <latexrelease>\IncludeInRelease{2021/11/15}%
412 <latexrelease> {\DeclareMathVersion}{local alphabets}%
413 <latexrelease>\def\DeclareMathVersion#1{%
414 \namedef{g_nfss_frozen_mv_#1_t1}{}%
415 \expandafter\new@mathversion\csname mv@#1\endcsname}
416 <latexrelease>\EndIncludeInRelease

```

```

417 <latexrelease>\IncludeInRelease{0000/00/00}%
418 <latexrelease> {\DeclareMathVersion}{local alphabets}%
419 <latexrelease>\def\DeclareMathVersion#1{%
420 \expandafter\new@mathversion\csname mv@#1\endcsname}
421 <latexrelease>\EndIncludeInRelease

```

```

422 <*2ekernel>

```

(End of definition for \DeclareMathVersion.)

```
\new@mathversion
```

```
423 \def\new@mathversion#1{%
424 \expandafter\in@\expandafter#\expandafter{\version@list}%
425 \ifin@
426 \@font@info{Redeclaring math version
427 '\expandafter\gobblefour\string#1'}%
428 \else
429 \expandafter\newcount\csname c@\expandafter
430 \gobble\string#1\endcsname
431 \def\version@elt{\noexpand\version@elt\noexpand}%
432 \edef\version@list{\version@list\version@elt#1}%
433 \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.

```
434 \toks@{}%
435 \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.

```
436 \def\group@elt##1##2{%
437 \advance\count@\@ne
438 \addto@hook\toks@{\getanddefine@fonts##1##2}%
439 }%
440 \group@list
```

We set the counter for this math version to the number of math groups found in \group@list.

```
441 \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.

```
442 \def\alpha@elt##1##2##3{%
443 \ifx##2\no@alphabet@error
444 \toks@{\expandafter{\the\toks@\install@mathalphabet##1%
445 {\no@alphabet@error##1}}}
446 \else
447 \toks@{\expandafter{\the\toks@\install@mathalphabet##1%
448 {\select@group##1##2##3}}}
449 \fi
450 }%
451 \alpha@list
```

Finally we define the math version to expand to the contents of \toks@.

```
452 \xdef#1{\the\toks@}%
453 }
454 \onlypreamble\new@mathversion
```

(End of definition for \new@mathversion.)

\DeclareSymbolFont First drop any surplus m from the series argument then do what has been done since 1994.

```
455 </2ekernel>
456 <*2ekernel | latexrelease>
```

```

457 〈\latexrelease〉\IncludeInRelease{2022/11/01}%
458 〈\latexrelease〉 {\DeclareSymbolFont}{maybe drop m}%
459 \def\DeclareSymbolFont #1#2#3#4#5{%
460 \def\reserved@a{\DeclareSymbolFont@m@dropped{#1}{#2}{#3}}%
461 \edef\reserved@b{#4}%
462 \series@maybe@drop@one@m\reserved@b\reserved@b
463 \expandafter\reserved@a\expandafter{\reserved@b}{#5}%
464 }
465 \def\DeclareSymbolFont@m@dropped #1#2#3#4#5{%
466 \tempswafalse
467 \edef\reserved@b{#2}%
468 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
469 \ifx\reserved@b\reserved@c \tempswatrue\fi}%
470 \cdp@list
471 \if@tempswa
472 \ifundefined{sym#1}%
473 \ifnum\count18<15 %
474 \expandafter\new@mathgroup\csname sym#1\endcsname
475 \expandafter\new@symbolfont\csname sym#1\endcsname
476 {#2}{#3}{#4}{#5}%
477 \else
478 \@latex@error{Too many symbol fonts declared}\@eha
479 \fi
480 }%
481 }%
482 \font@info{Redeclaring symbol font '#1'}%

```

Update the group list.

```

483 \def\group@elt##1##2{%
484 \noexpand\group@elt\noexpand##1%
485 \expandafter\ifx\csname sym#1\endcsname##1%
486 \expandafter\noexpand\csname##2/#3/#4/#5\endcsname
487 \else
488 \noexpand##2%
489 \fi}%
490 \xdef\group@list{\group@list}%

```

Update the version list.

```

491 \def\version@elt##1{%
492 \expandafter
493 \SetSymbolFont@\expandafter##1\csname##2/#3/#4/#5\expandafter
494 \endcsname \csname sym#1\endcsname
495 }%
496 \version@list
497 }%
498 \else
499 \@latex@error{Encoding scheme '#2' unknown}\@eha
500 \fi
501 }%
502 \onlypreamble\DeclareSymbolFont
503 (/2ekernel | \latexrelease)
504 〈\latexrelease〉\EndIncludeInRelease
505 〈\latexrelease〉\IncludeInRelease{0000/00/00}%

```

```

506 ⟨latexrelease⟩ {\DeclareSymbolFont}{maybe drop m}%
507 ⟨latexrelease⟩
508 ⟨latexrelease⟩\let\DeclareSymbolFont\DeclareSymbolFont@m@dropped
509 ⟨latexrelease⟩\let\DeclareSymbolFont@m@dropped\@undefined
510 ⟨latexrelease⟩
511 ⟨latexrelease⟩\EndIncludeInRelease
512 {*2ekernel}

(End of definition for \DeclareSymbolFont.)
```

\group@list

```

513 \let\group@list\empty
514 \@onlypreamble\group@list

(End of definition for \group@list.)
```

\group@elt

```

515 \let\group@elt\relax
516 \@onlypreamble\group@elt

(End of definition for \group@elt.)
```

\new@symbolfont

```

517 \def\new@symbolfont#1#2#3#4#5{%
518 \toks@\expandafter{\group@list}%
519 \edef\group@list{\the\toks@\noexpand\group@elt\noexpand#1%
520 \expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
521 \def\version@elt##1{\toks@\expandafter{##1}%
522 \edef##1{\the\toks@\noexpand\getanddefine@fonts
523 #1\expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
524 \global\advance\csname c@\expandafter
525 \@gobble\string##1\endcsname\@ne
526 }%
527 \version@list
528 }
529 \@onlypreamble\new@symbolfont

(End of definition for \new@symbolfont.)
```

\SetSymbolFont First drop any surplus m from the series argument then do what has been done since 1994.

```

530 ⟨/2ekernel⟩
531 ⟨*2ekernel | latexrelease⟩
532 ⟨latexrelease⟩\IncludeInRelease{2022/11/01}%
533 ⟨latexrelease⟩ {\SetSymbolFont}{maybe drop m}%
534 \def\SetSymbolFont #1#2#3#4#5#6{%
535 \def\reserved@a{\SetSymbolFont@m@dropped{#1}{#2}{#3}{#4}}%
536 \edef\reserved@b{#5}%
537 \series@maybe@drop@one@m\reserved@b\reserved@b
538 \expandafter\reserved@a\expandafter{\reserved@b}{#6}%
539 }
```

```

540 \def\SetSymbolFont@m@dropped#1#2#3#4#5#6{%
541 \tempswafalse
542 \edef\reserved@b{#3}%
543 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
544 \ifx\reserved@b\reserved@c \tempswatrue\fi}%
545 \cdp@list
546 \if@tempswa
547 \expandafter\SetSymbolFont@
548 \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
549 \endcsname \csname sym#1\endcsname
550 \else
551 \@latex@error{Encoding scheme '#3' unknown}\@eha
552 \fi
553 }
554 \only\SetSymbolFont
555 </2ekernel | latexrelease>
556 <latexrelease>\EndIncludeInRelease
557 <latexrelease>\IncludeInRelease{0000/00/00}%
558 <latexrelease> {\SetSymbolFont}{maybe drop m}%
559 <latexrelease>
560 <latexrelease>\let\SetSymbolFont\SetSymbolFont@m@dropped
561 <latexrelease>\let\SetSymbolFont@m@dropped@\undefined
562 <latexrelease>
563 <latexrelease>\EndIncludeInRelease
564 <*2ekernel>

```

(End of definition for \SetSymbolFont.)

```

\SetSymbolFont@
565 \def\SetSymbolFont@#1#2#3{%
566 \expandafter\in@\expandafter#1\expandafter{\version@list}%
567 \ifin@
568 \expandafter\in@\expandafter#3\expandafter{\group@list}%
569 \ifin@
570 \begingroup
571 \expandafter\get@cdp\string#2\@nil\reserved@a
572 \toks@{}%
573 \def\install@mathalphabet##1##2{%
574 \addto@hook\toks@{\install@mathalphabet##1##2}%
575 }%
576 \def\getanddefine@fonts##1##2{%
577 \ifnum##1=##2%
578 \addto@hook\toks@{\getanddefine@fonts##2}%
579 \expandafter\get@cdp\string##2\@nil\reserved@b
580 \ifx\reserved@a\reserved@b\else
581 \font@info{Encoding '\reserved@b' has changed
582 to '\reserved@a' for symbol font\MessageBreak
583 '\expandafter\gobblefour\string#3' in the
584 math version '\expandafter
585 \gobblefour\string#1'}%
586 \fi
587 \font@info{%
588 Overwriting symbol font
589 '\expandafter\gobblefour\string#3' in

```

```

590 version '\expandafter
591 \@gobblefour\string#1'\MessageBreak
592 \@spaces \expandafter\@gobble\string##2 -->
593 \expandafter\@gobble\string#2}%
594 \else
595 \addto@hook{\toks@{\getanddefine@fonts##1##2}%
596 \fi}%
597 #1%
598 \xdef#1{\the\toks@}%
599 \endgroup
600 \else
601 \@latex@error{Symbol font '\expandafter\@gobblefour\string#3'
602 not defined}\@eha
603 \fi
604 \else
605 \@latex@error{Math version '\expandafter\@gobblefour\string#1'
606 is not
607 defined}{You probably misspelled the name of the math
608 version.^^JOr you have to specify an additional package.}%
609 \fi
610 }
611 \onlypreamble\SetSymbolFont@
```

(End of definition for \SetSymbolFont@.)

```
\get@cdp
612 \def\get@cdp#1#2/#3@nil#4{\def#4{#2}}
613 \onlypreamble\get@cdp
```

(End of definition for \get@cdp.)

### \DeclareMathAlphabet

```

614 \def\DeclareMathAlphabet#1#2#3#4#5{%
615 \tempswafalse
616 \edef\reserved@b{#2}%
617 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
618 \ifx\reserved@b\reserved@c \tempswatrue\fi}%
619 \cdp@list
620 \if@tempswa
621 \expandafter\ifx
622 \csname\expandafter\@gobble\string#1\endcsname
623 \relax
624 \new@mathalphabet#1{#2}{#3}{#4}{#5}%
625 \else
```

Check if it is already a math alphabet.

```

626 \edef\reserved@a{\noexpand\in@\{\string\select@group\}%
627 {\expandafter\meaning\csname \expandafter
628 \@gobble\string#1\space\endcsname}\}%
629 \reserved@a
630 \ifin@
631 \font@info{Redeclaring math alphabet \string#1}%
632 \def\version@elt##1{%
633 \expandafter\SetMathAlphabet@\\expandafter
634 ##1\csname#2/#3/#4/#5\expandafter\endcsname}
```

```

635 \csname M@#2\expandafter\endcsname
636 \csname \expandafter\gobble\string#1\space\endcsname#1}%
637 \version@list
638 \else

```

Check if it is a math alphabet defined via `\DeclareSymbolFontAlphabet`.

```

639 \edef\reserved@a{\noexpand\in@\{\string\use@mathgroup\}%
640 {\expandafter\meaning\csname \expandafter
641 \gobble\string#1\space\endcsname}\}%
642 \reserved@a
643 \ifin@

```

In that case overwriting is simple since there is nothing inserted in the math version macros.

```

644 \font@info{Redeclaring math alphabet \string#1}%
645 \new@mathalphabet#1{#2}{#3}{#4}{#5}%

```

Otherwise panic.

```

646 \else
647 \@latex@error{Command '\string#1' already defined}\@eha
648 \fi
649 \fi
650 \fi
651 \else
652 \@latex@error{Encoding scheme '#2' unknown}\@eha
653 \fi
654 }
655 \onlypreamble\DeclareMathAlphabet

```

*(End of definition for `\DeclareMathAlphabet`.)*

`\new@mathalphabet`

```

656 \def\new@mathalphabet#1#2#3#4#5{%
657 \toks@\expandafter{\alpha@list}%
658 \edef#1{\expandafter\noexpand\csname \expandafter
659 \gobble\string#1\space\endcsname
660 \if/#5/%
661 \noexpand\no@alphabet@error
662 \noexpand\no@alphabet@error
663 \else
664 \expandafter\noexpand\csname M@#2\endcsname
665 \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
666 \fi
667 }%
668 \toks2\expandafter{#1}%
669 \edef\alpha@list{\the\toks@\noexpand\alpha@elt\the\toks2}%
670 \def\version@elt##1{\toks@\expandafter{##1}%
671 \edef##1{\the\toks@\install@mathalphabet
672 \expandafter\noexpand
673 \csname \expandafter\gobble
674 \string#1\space\endcsname
675 \if/#5/%
676 \noexpand\no@alphabet@error
677 \noexpand#1%
678 \else

```

```

679 \noexpand\select@group\the\toks2
680 \fi}\}%
681 }%
682 \version@list
683 \expandafter\edef\csname \expandafter\@gobble
684 \string#1\space\endcsname{\if/#5/%
685 \noexpand\no@alphabet@error
686 \noexpand#1%
687 \else
688 \noexpand\select@group\the\toks2
689 \fi}\}%
690 \edef#1{\noexpand\protect
691 \expandafter\noexpand\csname \expandafter
692 \@gobble\string#1\space\endcsname}\}%
693 }
694 \onlypreamble\new@mathalphabet

```

(End of definition for `\new@mathalphabet`.)

### `\SetMathAlphabet`

```

695 \def\SetMathAlphabet#1#2#3#4#5#6{%
696 \tempswafalse
697 \edef\reserved@b{\#3}%
698 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
699 \ifx\reserved@b\reserved@c \tempswatrue\fi}%
700 \cdp@list
701 \if@tempswa
702 \expandafter\SetMathAlphabet@
703 \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
704 \endcsname \csname M@#3\expandafter\endcsname
705 \csname \expandafter\@gobble\string#1\space\endcsname#1%
706 \else
707 \latext@error{Encoding scheme '#3' unknown}\eha
708 \fi
709 }
710 \onlypreamble\SetMathAlphabet

```

(End of definition for `\SetMathAlphabet`.)

### `\SetMathAlphabet@`

```

711 \def\SetMathAlphabet@#1#2#3#4#5{%
712 \expandafter\in@\expandafter#1\expandafter{\version@list}\}%
713 \ifin@
714 \expandafter\in@\expandafter#4\expandafter{\alpha@list}\}%
715 \ifin@
716 \begingroup
717 \toks@{}\}%
718 \def\getanddefine@fonts##1##2{%
719 \addto@hook\toks@{\getanddefine@fonts##1##2}\}%
720 }%
721 \def\reserved@c##1##2##3##4{%
722 \expandafter\@gobble\string##4}\}%
723 \def\install@mathalphabet##1##2{%
724 \ifx##1##2%
725 \addto@hook\toks@

```

```

726 {\install@mathalphabet#4{\select@group#4#3#2}}%
727 @font@info{Overwriting math alphabet
728 'string#5' in version '\expandafter
729 \gobblefour\string#1'\MessageBreak
730 @spaces \reserved@c##2 -->
731 \expandafter\gobble\string#2}%
732 \else
733 \addto@hook\toks@{\install@mathalphabet##1{##2}}%
734 \fi
735 }%
736 #1%
737 \xdef#1{\the\toks@}%
738 \endgroup
739 \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.

```

740 \edef\reserved@a{%
741 \noexpand\in@\string\use@mathgroup}{\meaning#4}%
742 \reserved@a
743 \ifin@
744 \def\reserved@b##1\use@mathgroup##2##3{%
745 \def\reserved@b{##3}\def\reserved@c{##2}}%
746 \expandafter\reserved@b#4%
747 \begingroup
748 \def\install@mathalphabet##1##2{%
749 \addto@hook\toks@{\install@mathalphabet##1{##2}}%
750 }%
751 \def\getanddefine@fonts##1##2{%
752 \addto@hook\toks@{\getanddefine@fonts##1##2}}%
753 \ifnum##1=\reserved@b
754 \expandafter
755 \addto@hook\expandafter\toks@
756 \expandafter{\expandafter\install@mathalphabet
757 \expandafter#4\expandafter
758 {\expandafter\select@group\expandafter
759 #4\reserved@c##2}}%
760 \fi
761 }%
762 \def\version@elt##1{%
763 \toks@{}%
764 ##1%
765 \xdef##1{\the\toks@}%
766 }%
767 \version@list
768 \endgroup

```

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

```

769 \expandafter\gdef\expandafter\alpha@list\expandafter
770 {\alpha@list
771 \alpha@elt #4\no@alphabet@error \no@alphabet@error}%
772 \gdef#4{\no@alphabet@error #5}%

```

Then call the internal setting routine again:

```

773 \SetMathAlphabet@{\#1}{\#2}{\#3}#4#5%
774 \else
775 \@latex@error{Command '\string#5' not defined as a
776 math alphabet}%
777 {Use \noexpand\DeclareMathAlphabet to define it.}%
778 \fi
779 \fi
780 \else
781 \@latex@error{Math version '\expandafter\gobblefour\string#1'
782 is not
783 defined}{You probably misspelled the name of the math
784 version.^JOr you have to specify an additional package.}%
785 \fi
786 }
787 \onlypreamble\SetMathAlphabet@
```

(End of definition for \SetMathAlphabet@.)

\DeclareMathAccent Could do with more checks like allowing single number in #4 lowercase in #4 etc

```

788 </2ekernel>
789 <*2ekernel | latexrelease>
790 <latexrelease>\IncludeInRelease{2019/10/01}%
791 <latexrelease> {DeclareMathAccent}{Make math accents robust}%
792 \def\DeclareMathAccent#1#2#3#4{%
793 \expandafter\in@\csname sym#3\expandafter\endcsname
794 \expandafter{\group@list}%
795 \ifin@
796 \begingroup
797 \count\z@=#4\relax
798 \count\tw@\count\z@
799 \divide\count\z@\sixt@@n
800 \count@\count\z@
801 \multiply\count@\sixt@@n
802 \advance\count\tw@-\count@
803 \if\relax\noexpand#1% is command?
804 \edef\reserved@a{\noexpand\in@
805 {\expandafter\gobble\string\mathaccent}
806 {\expandafter\meaning
807 \csname\expandafter\gobble\string#1\space\endcsname}%
808 \reserved@a
809 \ifin@
810 \expandafter\let
811 \csname\expandafter\gobble\string#1\space\endcsname
812 \undefined
813 \expandafter\set@mathaccent
814 \csname sym#3\endcsname#1#2%
815 {\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
816 \Qfont@info{Redeclaring math accent \string#1}%
817 \else
818 \expandafter\ifx
819 \csname\expandafter\gobble\string#1\endcsname
820 \relax
821 \expandafter\set@mathaccent
822 \csname sym#3\endcsname#1#2%
```

```

823 {\hexnumber@{\count\z@\hexnumber@{\count\tw@}}\%
824 \else
825 \@latex@error{Command ‘\string#1’ already defined}\@eha
826 \fi
827 \fi
828 \else
829 \@latex@error{Not a command name: ‘\noexpand#1’}\@eha
830 \fi
831 \endgroup
832 \else
833 \@latex@error{Symbol font ‘#3’ is not defined}\@eha
834 \fi
835 }
836 (/2ekernel | latexrelease)
837 <latexrelease>\EndIncludeInRelease
838 <latexrelease>\IncludeInRelease{0000/00/00}%
839 <latexrelease> {DeclareMathAccent}{Make math accents robust}%
840 <latexrelease>\def\DeclareMathAccent#1#2#3#4{%
841 <latexrelease> \expandafter\in@{\csname sym#3\expandafter\endcsname
842 <latexrelease> \expandafter{\group@list}%
843 <latexrelease> \ifin@
844 <latexrelease> \begingroup
845 <latexrelease> \count\z@=\#4\relax
846 <latexrelease> \count\tw@\count\z@
847 <latexrelease> \divide\count\z@\sixt@@n
848 <latexrelease> \count@\count\z@
849 <latexrelease> \multiply\count@\sixt@@n
850 <latexrelease> \advance\count\tw@-\count@
851 <latexrelease> \if\relax\noexpand#1% is command?
852 <latexrelease> \edef\reserved@a{\noexpand\in@
853 <latexrelease> {\expandafter\@gobble\string\mathaccent}{\meaning#1}}%
854 <latexrelease> \reserved@a
855 <latexrelease> \ifin@
856 <latexrelease> \expandafter\set@mathaccent
857 <latexrelease> \csname sym#3\endcsname#1#2%
858 <latexrelease> {\hexnumber@{\count\z@\hexnumber@{\count\tw@}}\%
859 <latexrelease> \font@info{Redeclaring math accent \string#1}}%
860 <latexrelease> \else
861 <latexrelease> \expandafter\ifx
862 <latexrelease> \csname\expandafter\@gobble\string#1\endcsname
863 <latexrelease> \relax
864 <latexrelease> \expandafter\set@mathaccent
865 <latexrelease> \csname sym#3\endcsname#1#2%
866 <latexrelease> {\hexnumber@{\count\z@\hexnumber@{\count\tw@}}\%
867 <latexrelease> \else
868 <latexrelease> \@latex@error{Command ‘\string#1’ already defined}\@eha
869 <latexrelease> \fi
870 <latexrelease> \fi
871 <latexrelease> \else
872 <latexrelease> \@latex@error{Not a command name: ‘\noexpand#1’}\@eha
873 <latexrelease> \fi
874 <latexrelease> \endgroup
875 <latexrelease> \else
876 <latexrelease> \@latex@error{Symbol font ‘#3’ is not defined}\@eha

```

```

877 〈\latexrelease〉 \fi
878 〈\latexrelease〉}
879 〈\latexrelease〉\EndIncludeInRelease
880 〈*2ekernel〉
881 \onlypreamble\DeclareMathAccent
(End of definition for \DeclareMathAccent.)

```

```

\set@mathaccent
882 〈/2ekernel〉
883 〈*2ekernel | \latexrelease〉
884 〈\latexrelease〉\IncludeInRelease{2019/10/01}%
885 〈\latexrelease〉 \set@mathaccent\{makemath accents robust}\%
886 \def\set@mathaccent#1#2#3#4{%
887 \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}%
888 \MakeRobust#2%
889 }
890 \onlypreamble\set@mathaccent
891 〈/2ekernel | \latexrelease〉
892 〈\latexrelease〉\EndIncludeInRelease
893 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
894 〈\latexrelease〉 \set@mathaccent\{makemath accents robust}\%
895 〈\latexrelease〉
896 〈\latexrelease〉\def\set@mathaccent#1#2#3#4{%
897 \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}%
898 〈\latexrelease〉
899 〈\latexrelease〉\EndIncludeInRelease
900 〈*2ekernel〉

```

(End of definition for \set@mathaccent.)

### \DeclareMathSymbol

```

901 \def\DeclareMathSymbol#1#2#3#4{%
902 \expandafter\in@\csname sym#3\expandafter\endcsname
903 \expandafter{\group@list}%
904 \ifin@
905 \begingroup
906 \count\z@=#4\relax
907 \count\tw@\count\z@
908 \divide\count\z@\sixt@@n
909 \count@\count\z@
910 \multiply\count@\sixt@@n
911 \advance\count\tw@-\count@
912 \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.

```

913 \edef\reserved@b{\expandafter\noexpand
914 \csname\expandafter\@gobble\string#1\space\endcsname}%

```

Test both #1 and #1<sub>1</sub> for containing mathchar.

```

915 \edef\reserved@a
916 {\noexpand\in@\{\expandafter\@gobble\string\mathchar\}%
917 \{\meaning#1\expandafter\meaning\reserved@b\}\}%
918 \reserved@a

```

Drop #1<sub>↓</sub> in case it was defined before.

```
919 \global\expandafter\let\reserved@b\@undefined
920 \ifin@
921 \expandafter\set@mathsymbol
922 \csname sym#3\endcsname#1#2%
923 {\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
924 \font@info{Redeclaring math symbol \string#1}%
925 \else
926 \expandafter\ifx
927 \csname\expandafter\gobble\string#1\endcsname
928 \relax
929 \expandafter\set@mathsymbol
930 \csname sym#3\endcsname#1#2%
931 {\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
932 \else
933 \@latex@error{Command ‘\string#1’ already defined}\@eha
934 \fi
935 \fi
936 \else
937 \expandafter\set@mathchar
938 \csname sym#3\endcsname#1#2
939 {\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
940 \fi
941 \endgroup
942 \else
943 \@latex@error{Symbol font ‘#3’ is not defined}\@eha
944 \fi
945 }
946 \onlypreamble\DeclareMathSymbol
```

(End of definition for `\DeclareMathSymbol`.)

```
\set@mathchar
947 \def\set@mathchar#1#2#3#4{%
948 \global\mathcode‘#2=”\mathchar@type#3\hexnumber@#1#4\relax}
949 \onlypreamble\set@mathchar
```

(End of definition for `\set@mathchar`.)

```
\set@mathsymbol
950 \def\set@mathsymbol#1#2#3#4{%
951 \global\mathchardef#2”\mathchar@type#3\hexnumber@#1#4\relax}
952 \onlypreamble\set@mathsymbol
```

(End of definition for `\set@mathsymbol`.)

```
953 \% \def\mathsymbol#1#2#3#4{%
954 % \tempcnta=#3\relax
955 % \tempcntb\tempcnta
956 % \divide\tempcnta\sixt@n
957 % \count@\tempcnta
958 % \multiply\count@\sixt@n
959 % \advance\tempcntb-\count@
960 % \mathchar”\mathchar@type#1\hexnumber@#2%
961 % \hexnumber@\tempcnta\hexnumber@\tempcntb\relax}
```

```

962 %
963 %\def\DeclareMathAlphabetCharacter#1#2#3{%
964 % \DeclareMathSymbol{#1}7{#2}{#3}}

```

### \DeclareMathDelimiter

```

965 \def\DeclareMathDelimiter#1{%
966 \if\relax\noexpand#1%
967 \expandafter\@DeclareMathDelimiter
968 \else
969 \expandafter\@xxDeclareMathDelimiter
970 \fi
971 #1}
972 \onlypreamble\DeclareMathDelimiter

```

(End of definition for \DeclareMathDelimiter.)

**\@xxDeclareMathDelimiter** This macro checks if the second arg is a “math type” such as `\mathopen`. The undocumented original code didn’t use math types when the delimiter was a single letter. For this reason the coding is a bit strange as it tries to support the undocumented syntax for compatibility reasons.

```

973 \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!

```

974 \begingroup
975 \let\mathalpha\mathord
976 \ifnum7=\mathchar@type{#2}%
977 \endgroup

```

If this branch is taken we have old syntax (5 arguments).

```

978 \expandafter\@firstofone
979 \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.

```

980 \endgroup
981 \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.

```

982 \expandafter\@firstoftwo
983 \fi
984 {\@xDeclareMathDelimiter{#1}{#2}{#3}{#4}}
985 \onlypreamble\@xxDeclareMathDelimiter

```

(End of definition for \@xxDeclareMathDelimiter.)

### \@DeclareMathDelimiter

```

986 \def\@DeclareMathDelimiter#1#2#3#4#5#6{%
987 \expandafter\in@\csname sym#3\expandafter\endcsname
988 \expandafter{\group@list}%
989 \ifin@

```

```

990 \expandafter\in@{\csname sym#5\expandafter\endcsname
991 \expandafter{\group@list}%
992 \ifin@
993 \begingroup
994 \count\z@=#4\relax
995 \count\tw@\count\z@
996 \divide\count\z@\sixt@@n
997 \count@\count\z@
998 \multiply\count@\sixt@@n
999 \advance\count\tw@-\count@
1000 \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
1001 %
1002 \count\z@=#6\relax
1003 \count\tw@\count\z@
1004 \divide\count\z@\sixt@@n
1005 \count@\count\z@
1006 \multiply\count@\sixt@@n
1007 \advance\count\tw@-\count@
1008 \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
1009 %
1010 \edef\reserved@a{\noexpand\in@
1011 {\expandafter\@gobble\string\delimiter}{\meaning#1}}%
1012 \reserved@a
1013 \ifin@
1014 \expandafter\set@mathdelimiter
1015 \csname sym#3\expandafter\endcsname
1016 \csname sym#5\endcsname#1#2%
1017 \reserved@c\reserved@d
1018 \font@info{Redeclaring math delimiter \string#1}%
1019 \else
1020 \expandafter\ifx
1021 \csname\expandafter\@gobble\string#1\endcsname
1022 \relax
1023 \expandafter\set@mathdelimiter
1024 \csname sym#3\expandafter\endcsname
1025 \csname sym#5\endcsname#1#2%
1026 \reserved@c\reserved@d
1027 \else
1028 \@latex@error{Command '\string#1' already defined}\@eha
1029 \fi
1030 \fi
1031 \endgroup
1032 \else
1033 \@latex@error{Symbol font '#5' is not defined}\@eha
1034 \fi
1035 \else
1036 \@latex@error{Symbol font '#3' is not defined}\@eha
1037 \fi
1038 }
1039 \onlypreamble\@DeclareMathDelimiter

```

(End of definition for \@DeclareMathDelimiter.)

\@xDeclareMathDelimiter

```

1040 \def\@xDeclareMathDelimiter#1#2#3#4#5{%
1041 \expandafter\in@\csname sym#2\expandafter\endcsname
1042 \expandafter{\group@list}%
1043 \ifin@
1044 \expandafter\in@\csname sym#4\expandafter\endcsname
1045 \expandafter{\group@list}%
1046 \ifin@
1047 \begingroup
1048 \count\z@=#3\relax
1049 \count\tw@\count\z@
1050 \divide\count\z@\sixt@@n
1051 \count@\count\z@
1052 \multiply\count@\sixt@@n
1053 \advance\count\tw@-\count@
1054 \edef\reserved@c{\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
1055 %
1056 \count\z@=#5\relax
1057 \count\tw@\count\z@
1058 \divide\count\z@\sixt@@n
1059 \count@\count\z@
1060 \multiply\count@\sixt@@n
1061 \advance\count\tw@-\count@
1062 \edef\reserved@d{\hexnumber@\{\count\z@\}\hexnumber@\{\count\tw@\}}%
1063 \expandafter\set@mathdelimiter
1064 \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
1065 \reserved@c\reserved@d
1066 \endgroup
1067 \else
1068 \@latex@error{Symbol font ‘#4’ is not defined}\@eha
1069 \fi
1070 \else
1071 \@latex@error{Symbol font ‘#2’ is not defined}\@eha
1072 \fi
1073 }
1074 \onlypreamble\@xDeclareMathDelimiter

```

(End of definition for `\@xDeclareMathDelimiter`.)

`\set@mathdelimiter` We have to end the definition of a math delimiter like `\lfloor` with a space and not with `\relax` as we did before, because otherwise constructs involving `\abovewithdelims` will prematurely end (pr/1329)

```

1075 </2ekernel>
1076 <*2ekernel | latexrelease>
1077 <latexrelease>\IncludeInRelease{2019/10/01}%
1078 <latexrelease> {\set@mathdelimiter}{make delimiters robust}%
1079 \def\set@mathdelimiter#1#2#3#4#5#6{%

```

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

```

1080 \protected
1081 \xdef#3{\delim@type#4\hexnumber@#1#5%
1082 \hexnumber@#2#6 }%
1083 % \MakeRobust#3%
1084 }

```

```

1085 \Oonlypreamble\set@mathdelimiter
1086 </2ekernel | latexrelease>
1087 <latexrelease>\EndIncludeInRelease
1088 <latexrelease>\IncludeInRelease{0000/00/00}%
1089 <latexrelease> {\set@mathdelimiter}{make delimiters robust}%
1090 <latexrelease>
1091 <latexrelease>\def\set@mathdelimiter#1#2#3#4#5#6{%
1092 <latexrelease> \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
1093 <latexrelease> \hexnumber@#2#6 }
1094 <latexrelease>
1095 <latexrelease>\EndIncludeInRelease
1096 {*2ekernel}

```

(End of definition for `\set@mathdelimiter`.)

```
\set@@mathdelimiter
1097 \def\set@@mathdelimiter#1#2#3#4#5{%
1098 <global\delcode'3="\hexnumber@#1#4\hexnumber@#2#5\relax}
1099 \Oonlypreamble\set@@mathdelimiter
```

(End of definition for `\set@@mathdelimiter`.)

`\DeclareMathRadical`

```
1100 \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!

```

1101 \expandafter\ifx
1102 <csname\expandafter\@gobble\string#1\endcsname
1103 <\relax
1104 <\let#1\radical
1105 <\fi
1106 \edef\reserved@a{\noexpand\in@
1107 <\expandafter\@gobble\string\radical}{\meaning#1}%
1108 \reserved@a
1109 \ifin@
1110 <\expandafter\in@\csname sym#2\expandafter\endcsname
1111 <\expandafter{\group@list}%
1112 \ifin@
1113 <\expandafter\in@\csname sym#4\expandafter\endcsname
1114 <\expandafter{\group@list}%
1115 \ifin@
1116 <\begingroup
1117 <\count\z@=#3\relax
1118 <\count\tw@\count\z@
1119 <\divide\count\z@\sixt@n
1120 <\count@\count\z@
1121 <\multiply\count@\sixt@n
1122 <\advance\count\tw@-\count@
1123 <\edef\reserved@c{%
1124 <\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
1125 <\count\z@=#5\relax
1126 <\count\tw@\count\z@
1127 <\divide\count\z@\sixt@n
1128 <\count@\count\z@

```

```

1129 \multiply\count@{sixt@}{n}
1130 \advance\count@{tw@-\count@}
1131 \edef\reserved@d{%
1132 \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
Coded inline instead of using \set@mathradical
1133 %
1134 % \expandafter\set@mathradical
1135 % \csname sym#2\expandafter\endcsname
1136 % \csname sym#4\endcsname#1%
1137 % \reserved@c\reserved@d
1138 \xdef#1{\radical"\expandafter\hexnumber@
1139 \csname sym#2\endcsname\reserved@c
1140 \expandafter\hexnumber@
1141 \csname sym#4\endcsname\reserved@d
1142 \relax}%
1143 \endgroup
1144 \else
1145 \@latex@error{Symbol font '#4' is not defined}\@eha
1146 \fi
1147 \else
1148 \@latex@error{Symbol font '#2' is not defined}\@eha
1149 \fi
1150 \else
1151 \@latex@error{Command '\string#1' already defined}\@eha
1152 \fi
1153 \onlypreamble\DeclareMathRadical

```

(End of definition for \DeclareMathRadical.)

Definition below was wrong it contained \delimiter !

```

def\set@mathradical#1#2#3#4#5{%
 \xdef#3{\radical"\hexnumber@#1#4\hexnumber@#2#5\relax}}

```

\mathalpha just a dummy currently  
1154 \let\mathalpha\relax

(End of definition for \mathalpha.)

\mathchar@type

```

1155 \def\mathchar@type#1{%
1156 \ifodd #1\#1\else % is this non-negative number?
1157 \ifx#1\mathord 0\else
1158 \ifx#1\mathop 1\else
1159 \ifx#1\mathbin 2\else
1160 \ifx#1\mathrel 3\else
1161 \ifx#1\mathopen 4\else
1162 \ifx#1\mathclose 5\else
1163 \ifx#1\mathpunct 6\else
1164 7% % anything else is variable ord
1165 \fi
1166 \fi
1167 \fi
1168 \fi
1169 \fi

```

```

1170 \fi
1171 \fi
1172 \fi}
1173 \onlypreamble\mathchar@type

```

(End of definition for \mathchar@type.)

### \DeclareSymbolFontAlphabet

```

1174 \def\DeclareSymbolFontAlphabet#1#2{%
1175 \expandafter\DeclareSymbolFontAlphabet@
1176 \csname \expandafter\gobble\string#1\space\endcsname{#2}#1}
1177 \onlypreamble\DeclareSymbolFontAlphabet

```

(End of definition for \DeclareSymbolFontAlphabet.)

### \DeclareSymbolFontAlphabet@

```

1178 \def\DeclareSymbolFontAlphabet@#1#2#3{%

```

We use the switch \if@tempswa to decide if we can declare this symbol font alphabet.

```

1179 \if@tempswa true

```

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

```

1180 \expandafter\in@\csname sym#2\expandafter\endcsname
1181 \expandafter{\group@list}%
1182 \ifin@
```

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

```

1183 \expandafter\in@\expandafter#1\expandafter{\alpha@list}%
1184 \ifin@
```

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

```

1185 @font@info{Redeclaring math alphabet \string#3}%
1186 \toks@{}%
1187 \def\alpha@elt##1##2##3{%
1188 \ifx##1#1\else\addto@hook\toks@{\alpha@elt##1##2##3}\fi}%
1189 \alpha@list
1190 \xdef\alpha@list{\the\toks@}%

```

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

```

1191 \def\version@elt##1{%
1192 \begingroup
1193 \toks@{}%
1194 \def\getanddefine@fonts####1####2{%
1195 \addto@hook\toks@{\getanddefine@fonts####1####2}}%
1196 \def\install@mathalphabet####1####2{%
1197 \ifx####1#1\else
1198 \addto@hook\toks@{\install@mathalphabet
1199 ####1{####2}}\fi}%
1200 ##1%
1201 \xdef##1{\the\toks@}%
1202 \endgroup
1203 }%
1204 \version@list
1205 \else

```

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

```
1206 \expandafter\ifx
1207 \csname\expandafter\gobble\string#1\space\endcsname
1208 \relax
```

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

```
1209 \else
1210 \edef\reserved@a{%
1211 \noexpand\in@\{\string\use@mathgroup\{\meaning#1}\}%
1212 \reserved@a
1213 \ifin@
1214 \font@info{Redeclaring math alphabet \string#3}%
1215 \else
```

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

```
1216 \tempswafalse
1217 \latex@error{Command '\string#3' already defined}\@eha
1218 \fi
1219 \fi
1220 \fi
1221 \else
```

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

```
1222 \tempswafalse
1223 \latex@error{Unknown symbol font '#2'}\@eha
1224 \fi
1225 \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.

```
1226 \def\group@elt##1##2{%
1227 \expandafter\ifx\csname sym#2\endcsname##1%
1228 \expandafter\reserved@a\string##2\@nil
1229 \fi}%
1230 \def\reserved@a##1##2##3\@nil{%
1231 \def\reserved@a{##2}%
1232 \group@list
1233 \toks@{\relax\ifmmode \else \non@alpherr#1\fi}%
1234 \edef#1{\the\toks@
1235 \noexpand\use@mathgroup
1236 \expandafter\noexpand\csname M@\reserved@a\endcsname
1237 \csname sym#2\endcsname}%
1238 \def#3{\protect#1}%
1239 \fi
1240 }
1241 \onlypreamble\DeclareSymbolFontAlphabet@
1242 </2ekernel>
```

*(End of definition for \DeclareSymbolFontAlphabet@.)*

## File 29

# ltfssini.dtx

This file contains the top level L<sup>A</sup>T<sub>E</sub>X interface to the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

## 1 NFSS Initialization

Finally, there are six commands that are to be used in L<sup>A</sup>T<sub>E</sub>X and that we will therefore protect against expansion at the wrong point: \fontfamily, \fontseries, \fontshape, \fontsize, \selectfont, and \mathversion.

```
1 {*2ekernel}
```

### 1.1 Providing math *versions*

L<sup>A</sup>T<sub>E</sub>X provides two *versions*. We call them *normal* and *bold*, respectively.

```
2 \DeclareMathVersion{normal}
3 \DeclareMathVersion{bold}
```

Now we define the standard font change commands. We don't allow the use of \rmfamily etc. in math mode.

(Actually most are now defined further down in the file.)

First the changes to another *family*:

```
4 \%{\ DeclareRobustCommand{\rmfamily
5 % {\not@math@alphabet\rmfamily\mathrm
6 % \fontfamily\rmdefault\selectfont}
7 \%{\ DeclareRobustCommand{\sffamily
8 % {\not@math@alphabet\sffamily\mathsf
9 % \fontfamily\sfdefault\selectfont}
10 \%{\ DeclareRobustCommand{\ttfamily
11 % {\not@math@alphabet\ttfamily\mathtt
12 % \fontfamily\ttdefault\selectfont}
```

Then the commands changing the *series*:

```
13 \%{\ DeclareRobustCommand{\bfseries
14 % {\not@math@alphabet\bfseries\mathbf
15 % \fontseries\bfdefault\selectfont}
16 \%{\ DeclareRobustCommand{\mdseries
17 % {\not@math@alphabet\mdseries\relax
18 % \fontseries\mddefault\selectfont}
19 \%{\ DeclareRobustCommand{\upshape
20 % {\not@math@alphabet\upshape\relax
21 % \fontshape\updefault\selectfont}
```

Then the commands changing the *shape*:

```
22 \%{\ DeclareRobustCommand{\slshape
23 % {\not@math@alphabet\slshape\relax
24 % \fontshape\sldefault\selectfont}
25 \%{\ DeclareRobustCommand{\scshape
26 % {\not@math@alphabet\scshape\relax
```

```

27 \fontshape\scdefault\selectfont}
28 \DeclareRobustCommand\itshape
29 {\not@math@alphabet\itshape\mathit
30 \fontshape\itdefault\selectfont}

```

## 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).

```

31 </2ekernel>
32 <*2ekernel | latexrelease>
33 <latexrelease>\IncludeInRelease{2021/11/15}%
34 <latexrelease> {\DeclareFontSeriesDefault}{Custom series}%

```

One problem with the NFSS approach of handling the series axis turned out to be that (especially with respect to “boldness”) different font families implemented different strategies. For example, with Computer Modern fonts you normally only have `bx` whereas most PostScript fonts offered only `b` but not `bx`. As a result L<sup>A</sup>T<sub>E</sub>X’s standard setting for `\bfdefault` didn’t work with such fonts, but if it got changed to produce `b`, then that didn’t work with Computer Modern if the fonts got combined (e.g., using Computer Modern Typewriter with such fonts).

The solution back then was to provide substitution rules in the font .fd such that if a `bx` series got requested the `b` series got used. While this works in that particular case, it isn’t a very general solution. For example, if you happen to have a font family that has several weights you may want to typeset the whole document in a somewhat lighter or darker font but if you then modify `\mddefault` to allow for this, then of course your change only works with that particular family but not with the typewriter or sans serif family you also want to use.

A better solution was provided by the `mweights` package by Bob Tennent that offers defaults on the level of the three main font families in the document: for “rm”, “sf” and “tt” so that font packages could define defaults for the sans serif document font by providing `\bfseries@sf` which then was used when `\bfseries` got executed and the current family was the `\sffamily`.

`\DeclareFontSeriesDefault`

We now support this concept directly from within L<sup>A</sup>T<sub>E</sub>X 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!<sup>36</sup>

`\DeclareFontSeriesDefault` The command to declare font series defaults for the “rm”, “sf” or “tt” family.

```

35 \let\DeclareFontSeriesDefault\@undefined % for rollback
36 \newcommand\DeclareFontSeriesDefault[3][]{%
37 \expandafter\def\@empty
38 \ifcsname#2series\endcsname % supported are
39 \ifcsname#3\endcsname % \[md/bf] default
40 \def\reserved@a{\#1}%

```

No optional argument: set up general default.

```

41 \ifx\reserved@a\@empty
42 \ifcsname #2series\endcsname % supported are
43 \ifcsname #3\endcsname % \[md/bf] default

```

Adding `\@empty` allows us to detect if the default gets redefined with `\renewcommand` or `\def` by the user.

```

44 \expandafter\def
45 \csname #2default\endcsname{\#3\@empty}%
46 \expandafter\def
47 \csname #2default@previous\endcsname{\#3\@empty}%
48 \else
49 \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
50 {Mandatory first argument must be 'md' or 'bf'.}
51 \fi

```

Optional argument given, set up specific default.

```

52 \else
53 \ifcsname #2series@\#1\endcsname % supported are
54 \ifcsname #3\endcsname % \[md/bf] series@[rm/sf/tt]
55 \expandafter\edef
56 \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).

```

57 \expandafter\let
58 \csname #2series@\#1@kernel\endcsname\@undefined
59 \else
60 \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
61 {Optional argument must be 'rm', 'sf', or 'tt'. \MessageBreak
62 Mandatory first argument must be 'md' or 'bf'.}
63 \fi
64 \fi
65 }

```

---

<sup>36</sup>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.

(End of definition for \DeclareFontSeriesDefault.)

```
66 {/2ekernel | latexrelease}
67 <latexrelease>\EndIncludeInRelease
68 <latexrelease>\IncludeInRelease{2020/02/02}%
69 <latexrelease> {\DeclareFontSeriesDefault}{Custom series}%
70 <latexrelease>
71 <latexrelease>\let\DeclareFontSeriesDefault\@undefined % for rollback
72 <latexrelease>\newcommand\DeclareFontSeriesDefault[3] []{%
73 <latexrelease> \def\reserved@a{\#1}%
74 <latexrelease> \ifx\reserved@a\empty%
75 <latexrelease> \ifcsname #2series\endcsname % supported are
76 <latexrelease> % \[md/bf]default
77 <latexrelease> \expandafter\def
78 <latexrelease> \csname #2default\endcsname{\#3\empty}%
79 <latexrelease> \expandafter\def
80 <latexrelease> \csname #2default@previous\endcsname{\#3\empty}%
81 <latexrelease> \else
82 <latexrelease> \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
83 <latexrelease> {Mandatory first argument must be 'md' or 'bf'.}
84 <latexrelease> \fi
85 <latexrelease> \else
86 <latexrelease> \ifcsname #2series@#1\endcsname % supported are
87 <latexrelease> % \[md/bf]series@[rm/sf/tt]
88 <latexrelease> \expandafter\edef
89 <latexrelease> \csname #2series@#1\endcsname{\#3}%
90 <latexrelease> \expandafter\let
91 <latexrelease> \csname #2series@#1@kernel\endcsname\@undefined
92 <latexrelease> \else
93 <latexrelease> \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
94 <latexrelease> {Optional argument must be 'rm', 'sf', or 'tt'. \MessageBreak
95 <latexrelease> Mandatory first argument must be 'md' or 'bf'.}
96 <latexrelease> \fi
97 <latexrelease> \fi
98 <latexrelease> }
99 <latexrelease>
100 <latexrelease>\EndIncludeInRelease
101 <latexrelease>\IncludeInRelease{0000/00/00}%
102 <latexrelease> {\DeclareFontSeriesDefault}{Custom series}%
103 <latexrelease>
104 <latexrelease>\let\DeclareFontSeriesDefault\@undefined
105 <latexrelease>\let\bfseries@rm\@undefined
106 <latexrelease>\let\bfseries@sf\@undefined
107 <latexrelease>\let\bfseries@tt\@undefined
108 <latexrelease>\let\bfseries@rm@kernel\@undefined
109 <latexrelease>\let\bfseries@sf@kernel\@undefined
110 <latexrelease>\let\bfseries@tt@kernel\@undefined
111 <latexrelease>\let\mdseries@rm\@undefined
112 <latexrelease>\let\mdseries@sf\@undefined
113 <latexrelease>\let\mdseries@tt\@undefined
114 <latexrelease>\expandafter\let\csname ver@mweights.sty\endcsname\@undefined
115 <latexrelease>
116 <latexrelease>\let\@meta@family@list\@undefined
117 <latexrelease>\let\prepare@family@series@update\@undefined
118 <latexrelease>\let@update@series@target@value\@undefined
```

```
119 \begin{macro}{\textrm{}}
```

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

```
120 \begin{macro}{\textrm}\let\init@series@setup\relax
121 \begin{macro}{\textrm}
122 \begin{macro}{\textrm}\EndIncludeInRelease
123 \begin{macro}{\textrm}{*2ekernel}
124 \begin{macro}{\textrm}{/2ekernel}
125 \begin{macro}{\textrm}{*2ekernel | \textrm{}}{2020/02/02}
126 \begin{macro}{\textrm}{\textrm{}}{mdseries@rm}{Custom series}
```

`\mdseries@rm` We initialize the family specific default at the end of the format generation. Later on they may get overwritten in the preamble or a package via `\DeclareFontSeriesDefault` (or possibly directly).

`\bfseries@rm` Conceptual change: The `\bfdefault` will be `b` not `bx` because that is what it should be really for nearly every font except Computer/Latin Modern.

`\bfseries@sf` 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.

```
128 \def\bfs@rm{bx}
129 \def\bfs@sf{bx}
130 \def\bfs@tt{bx}
```

Frozen version of the kernel defaults so we can see if they have changed.

```
131 \let\bfs@rm@kernel\bfs@rm
132 \let\bfs@sf@kernel\bfs@sf
133 \let\bfs@tt@kernel\bfs@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.

```
134 \def\mdseries@rm{m}
135 \def\mdseries@sf{m}
136 \def\mdseries@tt{m}
```

(*End of definition for `\mdseries@rm` and others.*)

`\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.

```
137 \begin{macro}{*debug}
138 \begin{macro}{\series@change@debug}\typeout
139 \begin{macro}{\series@change@debug}\@gobble
140 \begin{macro}{\series@change@debug}
```

(*End of definition for `\series@change@debug`.*)

`\prepare@family@series@update` This is core command that prepares for the family update. The big difference to the documented code above is that the nested `\ifx` statements seem to be missing. Instead we loop through an internal list that holds the names of the three meta families. This approach allows us to extend the mechanism at a later stage to allow for additional named meta families.

Here is the current definition of that list:

```
141 \def\@meta@family@list{\@elt{rm}\@elt{sf}\@elt{tt}}
```

```
142 \def\prepare@family@series@update#1#2{%
```

```
143 \if@forced@series
```

```
144 <+debug> \series@change@debug{No series preparation (forced \f@series)\on@line}%
145 \fontfamily#2%
```

```
146 \else
```

```
147 <+debug> \series@change@debug{Preparing for switching to #1 (#2)\on@line}%
148 \expand@font@defaults
```

We prepare for changing the current series. We have to find it before changing the family as discussed above.

```
149 \let\target@series@value\empty
150 \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}\allowbreak\verb`v'`
- 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.

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

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

```
152 \let\@elt\update@series@target@value
153 \@meta@family@list
```

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

```
154 \@elt{??}%
155 \let\@elt\relax
```

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

```
156 \fontfamily#2%
```

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

```
157 \ifx\target@series@value\empty
158 <+debug> \series@change@debug{Target series still empty ...}%
159 \else
160 \ifx \f@series\target@series@value
161 <+debug> \series@change@debug{Target series unchanged:
162 <+debug> \f@series \space = \target@series@value}%
163 \else
164 \maybe@load@fontshape
165 <+debug> \series@change@debug{Target series:
166 <+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 assign it to `\f@series` because we have to drop any surplus `m` first.

```

167 % \let\f@series\target@series@value
168 \series@maybe@drop@one@m\target@series@value\f@series
169 \fi
170 \fi
171 \fi
172 }
```

(End of definition for `\prepare@family@series@update` and `\@meta@family@list`.)

#### `\update@series@target@value`

In this macro used in the loop you basically find the nested `\ifxs` from the outline above. The only difference is that it is parameterized instead of being written out and only for one block of tests because the code is called repeatedly when looping over the meta family list. From the list we get each meta family name in turn.

```
173 \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.

```

174 \def\reserved@a{\#1}%
175 \ifx\target@meta@family@value\reserved@a % rm -> rm do nothing
176 \else
177 {+debug} \series@change@debug{Trying to match #1: \csname#1\def@ult\endcsname
178 {+debug} \space = \f@family\space ?}%

```

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

```
179 \expandafter\ifx\csname#1\def@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.

```
180 \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.

```

181 \expandafter\let\expandafter\reserved@b
182 \csname mdseries@\target@meta@family@value\endcsname
183 \expandafter\let\expandafter\reserved@c
184 \csname bfseries@\target@meta@family@value\endcsname
185 {+debug} \series@change@debug{Targets for mdseries and bfseries:
186 {+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.

```

187 \expandafter\series@maybe@drop@one@m
188 \csname mdseries@#1\endcsname\reserved@d
189 \ifx\reserved@d\f@series
190 {+debug} \series@change@debug{mdseries@#1 matched -> \reserved@b}%
191 \let\target@series@value\reserved@b
192 \else

```

Again do some sanitizing.

```
193 \expandafter\series@maybe@drop@one@m
194 \csname bfseries@#1\endcsname\reserved@d
195 \ifx\reserved@d\f@series
196 {+debug} \series@change@debug{bfseries@#1 matched -> \reserved@c}%
197 \let\target@series@value\reserved@c
198 \else\ifx\f@series\mddef@ult \let\target@series@value\reserved@b
199 {+debug} \series@change@debug{mddef@ult matched -> \reserved@b}%
200 \else\ifx\f@series\bfdef@ult \let\target@series@value\reserved@c
201 {+debug} \series@change@debug{bfdef@ult matched -> \reserved@c}%
202 \fi\fi\fi\fi
203 \fi
204 \fi
205 }
```

(End of definition for \update@series@target@value.)

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

```
206 \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.

```
207 \ifx\bfseries@rm@kernel\bfseries@rm
208 \expandafter\in@\expandafter{\rmdefault}%
209 {cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmmt}%
210 \ifin@ \else \def\bfseries@rm{b}\fi\fi
```

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

```
211 \ifx\bfseries@sf@kernel\bfseries@sf
212 \expandafter\in@\expandafter{\sfdefault}%
213 {cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmmt}%
214 \ifin@ \else \def\bfseries@sf{b}\fi\fi
215 \ifx\bfseries@tt@kernel\bfseries@tt
216 \expandafter\in@\expandafter{\ttdefault}%
217 {cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmmt}%
218 \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 survive 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`.

```

219 \reset@font
220 \ifx\seriesdefault\seriesdefault@kernel
221 \mdseries
222 \let\seriesdefault\f@series
223 \fi
224 }%

```

(End of definition for `\init@series@setup`.)

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.

```

225 \expandafter\let\csname ver@mweights.sty\endcsname\fmtversion
226 {/2ekernel | latexrelease}
227 \langle latexrelease\rangle\EndIncludeInRelease
228 \langle latexrelease\rangle\IncludeInRelease{0000/00/00}%
229 \langle latexrelease\rangle\{\mdseries@rm}-{Custom series}\%
230 \langle latexrelease\rangle
231 \langle latexrelease\rangle\let\bfseries@rm@\undefined
232 \langle latexrelease\rangle\let\bfseries@sf@\undefined
233 \langle latexrelease\rangle\let\bfseries@tt@\undefined
234 \langle latexrelease\rangle\let\bfseries@rm@kernel@\undefined
235 \langle latexrelease\rangle\let\bfseries@sf@kernel@\undefined
236 \langle latexrelease\rangle\let\bfseries@tt@kernel@\undefined
237 \langle latexrelease\rangle\let\mdseries@rm@\undefined
238 \langle latexrelease\rangle\let\mdseries@sf@\undefined
239 \langle latexrelease\rangle\let\mdseries@tt@\undefined
240 \langle latexrelease\rangle\expandafter\let\csname ver@mweights.sty\endcsname@\undefined
241 \langle latexrelease\rangle
242 \langle latexrelease\rangle\let\@meta@family@list@\undefined
243 \langle latexrelease\rangle\let\prepare@family@series@update@\undefined
244 \langle latexrelease\rangle\let\update@series@target@value@\undefined
245 \langle latexrelease\rangle

```

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

```

246 \langle latexrelease\rangle\let\init@series@setup\relax
247 \langle latexrelease\rangle
248 \langle latexrelease\rangle\EndIncludeInRelease
249 {*2ekernel}

```

```

250 </2ekernel>
251 <*2ekernel | latexrelease>
252 <latexrelease>\IncludeInRelease{2021/11/15}%
253 <latexrelease> {\bfseries}{Custom series with hooks}%

```

\bfseries This document command switches to the bold series.

```

254 \DeclareRobustCommand\bfseries{%
255 \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.

```

256 \expand@font@defaults
257 \maybe@update@bfseries@defaults
258 \ifx\f@family\rmdef@ult \fontseries\bfseries@rm
259 \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
260 \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt

```

If not \bfdefault is used.

```

261 \else \fontseries\bfdefault
262 \fi\fi\fi

```

This hook in contrast is always executed.

```

263 \UseHook{bfseries}%
264 \selectfont
265 }

```

(End of definition for \bfseries.)

\maybe@update@bfseries@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.

```

266 \def\maybe@update@bfseries@defaults{%
267 \ifx\bfdefault\bfdefault@previous\else

```

We add \@empty and then let \bfdefault@previous to \bfdefault so that we can detect any further change.

```

268 \expandafter\def\expandafter\bfdefault
269 \expandafter{\bfdefault\@empty}%
270 \let\bfdefault@previous\bfdefault

```

And we reset the meta family defaults (\bfdef@ult is an expanded version of \bfdefault).

```

271 \let\bfseries@rm\bfdef@ult
272 \let\bfseries@sf\bfdef@ult
273 \let\bfseries@tt\bfdef@ult

```

Formats that set up parallel fonts, e.g., for Japanese, can use this hook to add resets here. Note that this hook is only run when resets are necessary.

```

274 \UseHook{bfseries/defaults}%
275 \fi
276 }

```

(End of definition for \maybe@update@bfseries@defaults.)

\mdseries This document command switches to the medium series.

```
277 \DeclareRobustCommand\mdseries{%
278 \not@math@alphabet\mdseries\relax
279 \expand@font@defaults
280 \maybe@update@\mdseries@defaults
281 \ifx\f@family\rmdef@ult \fontseries\mdseries@rm
282 \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
283 \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
284 \else \fontseries\mddefault
285 \fi\fi\fi
286 \UseHook{\mdseries}%
287 \selectfont
288 }
```

(End of definition for \mdseries.)

\maybe@update@\mdseries@defaults

```
289 \def\maybe@update@\mdseries@defaults{%
290 \ifx\mddefault\mddefault@previous\else
291 \expandafter\def\expandafter\mddefault\expandafter{\mddefault\empty}%
292 \let\mddefault@previous\mddefault
293 \let\mdseries@rm\mddef@ult
294 \let\mdseries@sf\mddef@ult
295 \let\mdseries@tt\mddef@ult
```

Formats that set up parallel fonts, e.g., for Japanese, can use this hook to add resets here.

```
296 \UseHook{\mdseries/defaults}%
297 \fi
298 }
```

(End of definition for \maybe@update@\mdseries@defaults.)

```
299 </2ekernel | latexrelease>
300 <latexrelease>\EndIncludeInRelease
301 <latexrelease>\IncludeInRelease{2020/10/01}%
302 <latexrelease> {\bfseries}{Custom series with hooks}%
303 <latexrelease>
304 <latexrelease>\let\maybe@update@bfseries@defaults\undefined
305 <latexrelease>\let\maybe@update@\mdseries@defaults\undefined
306 <latexrelease>
307 <latexrelease>\DeclareRobustCommand\bfseries{%
308 \not@math@alphabet\bfseries\mathbf
309 \expand@font@defaults
310 \ifx\bfdefault\bfdefault@previous\else
311 \expandafter\def\expandafter\bfdefault
312 \expandafter{\bfdefault\empty}%
313 \let\bfdefault@previous\bfdefault
314 \let\bfseries@rm\bfdef@ult
315 \let\bfseries@sf\bfdef@ult
316 \let\bfseries@tt\bfdef@ult
317 \UseHook{\bfseries/defaults}%
318 \fi
319 \ifx\f@family\rmdef@ult \fontseries\bfseries@rm
320 \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
```

```

321 〈latexrelease〉 \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt
322 〈latexrelease〉 \else \fontseries\bfdefault
323 〈latexrelease〉 \fi\fi\fi
324 〈latexrelease〉 \UseHook{bfseries}%
325 〈latexrelease〉 \selectfont
326 〈latexrelease〉}
327 〈latexrelease〉
328 〈latexrelease〉\DeclareRobustCommand\mdseries{%
329 〈latexrelease〉 \not@math@alphabet\mdseries\relax
330 〈latexrelease〉 \expand@font@defaults
331 〈latexrelease〉 \ifx\mddefault\mddefault@previous\else
332 〈latexrelease〉 \expandafter\def\expandafter\mddefault\expandafter{\mddefault\emptyset}%
333 〈latexrelease〉 \let\mddefault@previous\mddefault
334 〈latexrelease〉 \let\mdseries@rm\mddef@ult
335 〈latexrelease〉 \let\mdseries@sf\mddef@ult
336 〈latexrelease〉 \let\mdseries@tt\mddef@ult
337 〈latexrelease〉 \UseHook{mdseries/defaults}%
338 〈latexrelease〉 \fi
339 〈latexrelease〉 \ifx\f@family\rmdef@ult \fontseries\mdseries@rm
340 〈latexrelease〉 \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
341 〈latexrelease〉 \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
342 〈latexrelease〉 \else \fontseries\mddefault
343 〈latexrelease〉 \fi\fi\fi
344 〈latexrelease〉 \UseHook{mdseries}%
345 〈latexrelease〉 \selectfont
346 〈latexrelease〉}
347 〈latexrelease〉\EndIncludeInRelease
348 〈latexrelease〉\IncludeInRelease[2020/02/02]%
349 〈latexrelease〉 {\bfseries}{Custom series with hooks}%
350 〈latexrelease〉
351 〈latexrelease〉
352 〈latexrelease〉\DeclareRobustCommand\bfseries{%
353 〈latexrelease〉 \not@math@alphabet\bfseries\mathbf
354 〈latexrelease〉 \expand@font@defaults
355 〈latexrelease〉 \ifx\f@family\rmdef@ult \fontseries\bfseries@rm
356 〈latexrelease〉 \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
357 〈latexrelease〉 \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt
358 〈latexrelease〉 \else \fontseries\bfdefault
359 〈latexrelease〉 \fi\fi\fi
360 〈latexrelease〉 \selectfont
361 〈latexrelease〉}
362 〈latexrelease〉
363 〈latexrelease〉\DeclareRobustCommand\mdseries{%
364 〈latexrelease〉 \not@math@alphabet\mdseries\relax
365 〈latexrelease〉 \expand@font@defaults
366 〈latexrelease〉 \ifx\f@family\rmdef@ult \fontseries\mdseries@rm
367 〈latexrelease〉 \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
368 〈latexrelease〉 \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
369 〈latexrelease〉 \else \fontseries\mddefault
370 〈latexrelease〉 \fi\fi\fi
371 〈latexrelease〉 \selectfont
372 〈latexrelease〉}
373 〈latexrelease〉
374 〈latexrelease〉

```

```

375 ⟨latexrelease⟩
376 ⟨latexrelease⟩\EndIncludeInRelease
377 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
378 ⟨latexrelease⟩ {\bfseries}{Custom series with hooks}%
379 ⟨latexrelease⟩
380 ⟨latexrelease⟩\DeclareRobustCommand\bfseries
381 ⟨latexrelease⟩ {\not@math@alphabet\bfseries\mathbf}
382 ⟨latexrelease⟩ \fontseries\bfdefault\selectfont
383 ⟨latexrelease⟩\DeclareRobustCommand\mdseries
384 ⟨latexrelease⟩ {\not@math@alphabet\mdseries\relax
385 ⟨latexrelease⟩ \fontseries\mddefault\selectfont}
386 ⟨latexrelease⟩
387 ⟨latexrelease⟩\EndIncludeInRelease
388 {*2ekernel}
389
390
391
392
393 ⟨/2ekernel⟩
394 {*2ekernel | latexrelease}
395 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
396 ⟨latexrelease⟩ {\expand@font@defaults}{Custom series with hooks}%

```

`\expand@font@defaults` The family specific defaults are fully expanded, i.e., they are defined via `\edef` inside `\DeclareFontSeriesDefault`. However, the overall defaults, e.g., `\bfdefault` may have been redefined by the user and thus may not be fully expanded. So to enable reliable comparison we make expanded versions of them. That we rerun each time. The alternative would be to only allow for changes before begin document.

`\rm@def@ult` 397 `\def\expand@font@defaults{%`

`\sf@def@ult` 398 `\edef\rmdef@ult{\rmdefault}%`

`\tt@def@ult` 399 `\edef\sfdef@ult{\sfdefault}%`

`\md@def@ult` 400 `\edef\ttdef@ult{\ttdefault}%`

The series defaults may contain some surplus `m` that we need to drop here.

```

401 \series@maybe@drop@one@m\bfdefault\bfdef@ult
402 \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.

```

403 \UseHook{\expand@font@defaults}%
404 }

```

(End of definition for `\expand@font@defaults` and others.)

`\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.

```

\rmfamily\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 have different sets of shapes available and we certainly don't want to see substitution going on. So we use `\target@series@value` to hold the target series (if any).

```
\let\target@series@value\empty
```

Thus, if the current family is the sans family

```
\ifx\f@family\sfdef@ult
```

and if we using the medium series of the sans family

```
\ifx\f@series\mdseries@sf
```

then lets switch to the medium series for the serif family

```
\let\target@series@value\mdseries@rm
```

and if we use the bold series of the sans family switch to the bold default of the serif family:

```
\else\ifx\f@series\bfseries@sf \let\target@series@value\bfseries@rm
```

However, the sans family may not have any specific defaults set, so we also compare with the overall defaults.

```
\else\ifx\f@series\mddef@ult \let\target@series@value\mdseries@rm
\else\ifx\f@series\bfdef@ult \let\target@series@value\bfseries@rm
```

If neither test was true we leave the series alone. This way a special manual setting such as `\fontseries{lc}` is not undone if the family changes (of course there may not be any support for it in the new family but then the NFSS substitution kicks in and sorts it out).

```
\fi\fi\fi\fi
```

We need to do the same if the current family is the typewriter family:

```
\else\ifx\f@family\ttdef@ult
 \ifx\f@series\mdseries@tt \let\target@series@value\mdseries@rm
 \else\ifx\f@series\bfseries@tt \let\target@series@value\bfseries@rm
 \else\ifx\f@series\mddef@ult \let\target@series@value\mdseries@rm
 \else\ifx\f@series\bfdef@ult \let\target@series@value\bfseries@rm
 \fi\fi\fi\fi
\fi\fi
```

With these preparations for series out of the way we can now change the font family to `\rmdefault`.

```
\fontfamily\rmdefault
```

If `\target@series@value` is still empty there is nothing more to do other than selecting the new family. However, if not then we should update the font series now as well. But there is one further subtle issue. We may not have loaded an `.fd` file for our target font family yet. In the past that was done in `\selectfont` if necessary but since we are now doing all the comparisons in `\fontseries` we need to make sure that the font family specifications are already loaded prior to calling `\fontseries`.

```
\ifx\target@series@value\empty \else
 \maybe@load@fontshape
```

Updating the series in this case means directly changing `\f@series` to the target value. We don't want to go through `\fontseries` because that would apply the mappings and then `bx + b` would keep `bx` instead of changing to `b` as desired. as

```
\let\f@series\target@series@value
\fi
\selectfont}
```

So now for the real definition: most of the code above gets delegated to a helper command `\prepare@family@series@update` so that the definition becomes again fairly short. In addition we add a hook, mainly for our Japanese friends so that the code can be extended prior to the call to `\selectfont`.

```
405 \DeclareRobustCommand\rmfamily{%
 406 \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.

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

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

```
408 \UseHook{rmfamily}%
 409 \selectfont
```

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

```
\ttfamily 410 \DeclareRobustCommand\sffamily{%
 411 \not@math@alphabet\sffamily\mathsf
 412 \prepare@family@series@update{sf}\sfdefault
 413 \UseHook{sffamily}%
 414 \selectfont

 415 \DeclareRobustCommand\ttfamily{%
 416 \not@math@alphabet\ttfamily\mathtt
 417 \prepare@family@series@update{tt}\ttdefault
 418 \UseHook{ttfamily}%
 419 \selectfont}
```

*(End of definition for `\rmfamily`, `\sffamily`, and `\ttfamily`.)*

```
rmfamily Declare the hooks used above.
sffamily 420 \NewHook{rmfamily}
ttfamily 421 \NewHook{sffamily}
normalfont 422 \NewHook{ttfamily}
expand@font@defaults 423 \NewHook{normalfont}
bfseries 424 \NewHook{expand@font@defaults}
bfseries/defaults 425 \NewHook{bfseries}
mdseries 426 \NewHook{bfseries/defaults}
mdseries/defaults 427 \NewHook{mdseries}
428 \NewHook{mdseries/defaults}
```

(End of definition for `rmfamily` and others.)

`\@rmfamilyhook` These four hooks have legacy versions used in 2020/02/02 so we should support them  
`\@sffamilyhook` until they aren't any longer used.

By default the hooks do nothing.

```
\@defaultfamilyhook 429 \let\@rmfamilyhook\@empty
 430 \let\@sffamilyhook\@empty
 431 \let\@ttfamilyhook\@empty
 432 \let\@defaultfamilyhook\@empty %FMi sort out
```

(End of definition for \rmfamilyhook and others.)

```
433 </2ekernel | latexrelease>
434 <|latexrelease>\EndIncludeInRelease
435 <|latexrelease>\IncludeInRelease{2020/02/02}%
436 <|latexrelease> {\expandafter\font@defaults}{Custom series with hooks}%
437 <|latexrelease>
438 <|latexrelease>\def\expand@font@defaults{%
439 <|latexrelease> \edef\rmdef@ult{\rmdefault}%
440 <|latexrelease> \edef\sffdef@ult{\sfdefault}%
441 <|latexrelease> \edef\ttdef@ult{\ttdefault}%
442 <|latexrelease> \edef\bfdef@ult{\bfdefault}%
443 <|latexrelease> \edef\mddef@ult{\mddefault}%
444 <|latexrelease> \edef\famdef@ult{\familydefault}%
445 <|latexrelease>}
446 <|latexrelease>
447 <|latexrelease>
448 <|latexrelease>\DeclareRobustCommand\rmfamily{%
449 <|latexrelease> \not@math@alphabet\rmfamily\mathrm
450 <|latexrelease> \prepare@family@series@update{\rm}\rmdefault
451 <|latexrelease> \rmfamilyhook
452 <|latexrelease> \selectfont}
453 <|latexrelease>\DeclareRobustCommand\sffamily{%
454 <|latexrelease> \not@math@alphabet\sffamily\mathsf
455 <|latexrelease> \prepare@family@series@update{\sf}\sfdefault
456 <|latexrelease> \sffamilyhook
457 <|latexrelease> \selectfont}
458 <|latexrelease>\DeclareRobustCommand\ttfamily{%
459 <|latexrelease> \not@math@alphabet\ttfamily\mathtt
460 <|latexrelease> \prepare@family@series@update{\tt}\ttdefault
461 <|latexrelease> \ttfamilyhook
462 <|latexrelease> \selectfont}
463 <|latexrelease>\let@\rmfamilyhook\empty
464 <|latexrelease>\let@\sffamilyhook\empty
```

```

465 \let\@ttfamilyhook\@empty
466 \let\@empty
467 \EndIncludeInRelease
468 \IncludeInRelease{0000/00/00}%
469 {\@expand@font@defaults}{Custom series with hooks}%
470 \let\@expand@font@defaults\@undefined
471 \let\@empty
472 \let\@expand@font@defaults\@undefined
473 \let\@empty
474 \DeclareRobustCommand\bfs
475 {\not@math@alphabet\bfs\mathbf}
476 \fontseries\bfd\selectfont
477 \DeclareRobustCommand\mds
478 {\not@math@alphabet\mds\relax}
479 \fontseries\mdf\selectfont
480 \DeclareRobustCommand\rmf
481 {\not@math@alphabet\rmf\mathrm}
482 \fontfamily\rmf\selectfont
483 \DeclareRobustCommand\sff
484 {\not@math@alphabet\sff\mathsf}
485 \fontfamily\sff\selectfont
486 \DeclareRobustCommand\ttf
487 {\not@math@alphabet\ttf\mathtt}
488 \fontfamily\ttf\selectfont
489 \let\@empty
490 \let\@rmfamilyhook\@undefined
491 \let\@sffamilyhook\@undefined
492 \let\@ttfamilyhook\@undefined
493 \let\@empty
494 \EndIncludeInRelease
495 {*2ekernel}

```

\IfFontSeriesContextTF With the ability for `\bfs` or `\mds` 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 typesetting 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.,

```
\NewDocumentCommand\IfBold{mm}{\IfFontSeriesContextTF{bf}{#1}{#2}}
```

and then do

```
This is \IfBold{bold}{non-bold} text.
```

and get the appropriate result.

```
496 </2ekernel>
497 <*2ekernel | latexrelease>
498 <latexrelease>\IncludeInRelease{2020/10/01}%
499 <latexrelease> {\IfFontSeriesContextTF}{Font series context}%
500 \DeclareRobustCommand\IfFontSeriesContextTF[1]{%
501 \expand@font@defaults
```

In the beginning we haven't found the context we are looking for.

```
502 \@font@series@contextfalse
```

We store the requested context away for use in the tests.

```
503 \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.

```
504 \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 artificial meta family `??` and test each of them in turn using `\test@font@series@context` as the testing command.

```
505 \let\@elt\test@font@series@context
506 \cmeta@family@list
507 \celt{??}%
508 \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.

```
509 \if@font@series@context
510 \expandafter\@firstoftwo
511 \else
512 \expandafter\@secondoftwo
513 \fi
514 }
```

(End of definition for `\IfFontSeriesContextTF`.)

`\test@font@series@context` This tests the context (stored in `\requested@test@context`) and updates the boolean if the right context is found.

```
515 \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`.

```
516 \edef\reserved@a{\csname #1def@ult\endcsname}%
517 \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`.

```
518 \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).

```
519 \expandafter\ifx
520 \csname\requested@test@context series@\#1\endcsname\f@series
```

If yes we change the boolean and are done.

```
521 \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 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.

```
522 \else
523 \expandafter\ifx
524 \csname\requested@test@context def@ult\endcsname\f@series
525 \font@series@contexttrue
526 \fi\fi\fi
527 }
```

(End of definition for `\test@font@series@context`.)

`\if@font@series@context` The boolean to signal if we found the requested font series context.

```
528 \newif\if@font@series@context
```

(End of definition for `\if@font@series@context`.)

```
529 </2ekernel | latexrelease>
530 <latexrelease>\EndIncludeInRelease
531 <latexrelease>\IncludeInRelease{0000/00/00}%
532 <latexrelease> {\IfFontSeriesContextTF}{Font series context}%
533 <latexrelease>
534 <latexrelease>\let\IfFontSeriesContextTF\@undefined
535 <latexrelease>\let\test@font@series@context\@undefined
536 <latexrelease>\let\if@font@series@context\@undefined
537 <latexrelease>\let\@font@series@contexttrue\@undefined
538 <latexrelease>\let\@font@series@contextfalse\@undefined
539 <latexrelease>\EndIncludeInRelease
540 <*2ekernel>
```

### 3 Supporting nested emphasis

By default L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  supports two levels of nested emphasis: if the current font has an upright shape then it switches to `\itshape` otherwise to `\emnnershape` (which defaults to `\upshape`). This means nested emphasis will oscillate between italic and upright shapes.

Sometimes it would be nice to allow for a more lengthy sequence, but instead of providing a fixed one L<sup>A</sup>T<sub>E</sub>X 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 aclist of declaration. Spaces in the argument are dropped to avoid spurious 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.<sup>37</sup> Further nested calls restart at the beginning.

```

541 </2ekernel>
542 <*2ekernel | latexrelease>
543 <latexrelease> \IncludeInRelease{2020/02/02}%
544 <latexrelease> {\DeclareEmphSequence}{Nested emph}%
545 \def\DeclareEmphSequence#1{%
546 \protected@edef\emfontdeclare@clist{\zap@space#1, \empty\emforce\emreset}%
547 }

```

By default the it is empty, in which case `\eminnershape` is used by L<sup>A</sup>T<sub>E</sub>X.

```
548 \let\emfontdeclare@clist\empty
```

(End of definition for `\DeclareEmphSequence`.)

`\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:

```
549 \ DeclareRobustCommand\emrest{\upshape\ulcshape}
```

(End of definition for `\emrest`.)

`\em` The new definition for `\em` (and implicitly `\emph`) is the same as before as long as `\emfontdeclare@clist` is empty.

```

550 \ DeclareRobustCommand\em{%
551 \nomath\em
552 \ifx\emfontdeclare@clist\empty
553 \ifdim \fontdimen\ne\font >\z@
554 \eminnershape \else \itshape \fi
555 \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.

```
556 \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.

```

557 \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
558 \fi
559 }
```

---

<sup>37</sup> 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 ...

```
560 \def\eminnershape{\upshape}
```

(End of definition for \em.)

\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.

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

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

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

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

```
563 % \typeout{Use: \detokenize{#1}}%
```

```
564 #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.

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

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

```
566 \expandafter\endcsname
567 \em@currfont
```

```
568 \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
```

If \emforce was used, we have to undo its effect:

```
569 \else
570 \let\em@force\@empty
571 \fi
572 }
```

(End of definition for \do@emfont@update.)

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

```
573 \protected\def\emforce{\def\em@force{/}}
```

```
574 \let\em@force\@empty
```

```
575 </2ekernel | latexrelease>
```

```
576 <latexrelease>\EndIncludeInRelease
```

(End of definition for \emforce and \em@force.)

\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.

```
577 <latexrelease>\IncludeInRelease{2015/01/01}{\DeclareEmphSequence}{Nested emph}%
```

```
578 <latexrelease>\let\DeclareEmphSequence\@undefined
```

```
579 <latexrelease>\let\emfontdeclare@clist\@undefined
```

```
580 <latexrelease>\let\emreset\@undefined
```

```
581 <latexrelease>\let\do@emfont@update\@undefined
```

```

582 〈\latexrelease〉\let\emforce\@undefined
583 〈\latexrelease〉\let\em@force\@undefined
584 〈\latexrelease〉
585 〈\latexrelease〉\DeclareRobustCommand\em
586 〈\latexrelease〉 {‐@nomath\em \ifdim \fontdimen@ne\font >\z@
587 〈\latexrelease〉 \eminnershape \else \itshape \fi}%
588 〈\latexrelease〉\EndIncludeInRelease
589 〈\latexrelease〉
590 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\DeclareEmphSequence}{Nested emph}%
591 〈\latexrelease〉\DeclareRobustCommand\em
592 〈\latexrelease〉 {‐@nomath\em \ifdim \fontdimen@ne\font >\z@
593 〈\latexrelease〉 \upshape \else \itshape \fi}%
594 〈\latexrelease〉\let\eminnershape\@undefined
595 〈\latexrelease〉\EndIncludeInRelease
596 〈*2ekernel〉

```

(End of definition for `\em` and `\eminnershape`.)

`\not@math@alphabet` This function generates an error message when it is called in math mode. The same function should be defined in `newfont.sty`.

```

597 \def\not@math@alphabet#1#2{%
598 \relax
599 \ifmmode
600 \@latex@error{Command \noexpand#1 invalid in math mode}%
601 {%
602 Please
603 \ifx#2\relax
604 define a new math alphabet^{‐‐J}%
605 if you want to use a special font in math mode%
606 \else
607 use the math alphabet \noexpand#2 instead of
608 the #1 command%
609 \fi
610 .
611 }%
612 \fi}

```

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 :-).

```

607 use the math alphabet \noexpand#2 instead of
608 the #1 command%
609 \fi
610 .
611 }%
612 \fi}

```

(End of definition for `\not@math@alphabet`.)

Finally we provide two abbreviations to switch to the L<sup>A</sup>T<sub>E</sub>X versions.

```

613 \DeclareRobustCommand\boldmath{‐@nomath\boldmath
614 \mathversion{bold}}
615 \DeclareRobustCommand\unboldmath{‐@nomath\unboldmath
616 \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 `\gls@settings`.

```

617 \def\math@version{normal}

```

### 3.1 Legacy

We start by defining a few macros that are part of standard L<sup>A</sup>T<sub>E</sub>X'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
618 \def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
(End of definition for \newfont.)

\symbol
619 </2ekernel>
620 <*2ekernel | latexrelease>
621 <latexrelease>\IncludeInRelease{2020/10/01}%
622 <latexrelease> {\symbol}{XeTeX change for math}%
623 \ifdefined\XeTeXversion
624 \ DeclareRobustCommand\symbol[1]{\Ucharcat#1 12\relax}
625 \else
626 \ DeclareRobustCommand\symbol[1]{\char#1\relax}
627 \fi
628 </2ekernel | latexrelease>
629 <| latexrelease>\EndIncludeInRelease
630 <| latexrelease>\IncludeInRelease{0000/00/00}%
631 <| latexrelease> {\symbol}{XeTeX change for math}%
632 <| latexrelease>
633 <| latexrelease>\DeclareRobustCommand\symbol[1]{\char#1\relax}
634 <| latexrelease>
635 <| latexrelease>\EndIncludeInRelease
636 <*2ekernel>
(End of definition for \symbol.)
```

### 3.2 Miscellaneous

\@setfontsize This abbreviation is used by L<sup>A</sup>T<sub>E</sub>X's user level size changing commands, such as \large.  
\@setsizesize

```
637 \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.

```
638 \ifx\protect\@typeset@protect
639 \let\@currsize#1%
640 \fi
641 \fontsize{#2}{#3}\selectfont
```

For compatibility we also define \@setsizesize the 209 command

```
642 <*compat>
643 \def\@setsizesize#1#2#3#4{\@setfontsize#1{#4}{#2}}
644 </compat>
```

```
(End of definition for \@setfontsize and \@setsizesize.)
```

`\hexnumber@` To set up L<sup>A</sup>T<sub>E</sub>X's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple `\ifcase`.

```
645 \def\hexnumber@#1{\ifcase\number#1
646 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
647 9\or A\or B\or C\or D\or E\or F\fi}
```

(End of definition for `\hexnumber@`.)

`\nfss@text` In its simplest form `\nfss@text` is an `\mbox`. This will produce unbreakable text outside math and inside math you will get text with the same fonts as outside. The only drawback is that such item won't change sizes in subscripts. But this behavior can be easily changed. With the `amstex` style option one will get a sub style called `amstext` which will redefine the `\nfss@text` macro to produce correct text in all sizes.

We have to use `\def` instead of the shorter `\let` since `\mbox` is undefined when we reach this point.

```
648 \def\nfss@text#1{{\mbox{#1}}}
```

(End of definition for `\nfss@text`.)

`\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`.

```
649 %\DeclareRobustCommand\copyright
650 % {\oalign{\hfil
651 % \raise.07ex\hbox{\mdseries\upshape c}\hfil\crcr
652 % \mathhexbox20D}}
```

(End of definition for `\copyright`.)

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

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

```
653 /2ekernel)
654 (*2ekernel | latexrelease)
655 <latexrelease>\IncludeInRelease{2021/06/01}%
656 <latexrelease> {\normalfont}{Add hook to \normalfont}%
657 \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.

```
658 \fontencoding\encodingdefault
659 \edef\f@family{\familydefault}%
660 \edef\f@series{\seriesdefault}%
661 \edef\f@shape{\shapedefault}%
```

Any earlier `\fontseries`, etc. should be canceled and we should switch unconditionally to the requested font face so we drop any code that may have been stored in `\delayed@f@adjustment`.

```
662 \let\delayed@f@adjustment\empty
663 \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!

```
664 \@defaultfamilyhook % hookname from 2020/02 will vanish
665 \selectfont}
666 \let\reset@font\normalfont
(End of definition for \normalfont and \reset@font.)

667 % \changes{v3.2g}{2021/03/18}
668 % {Add missing 2020/02/02 latexrelease entry.}
669 </2ekernel | latexrelease>
670 <latexrelease>\EndIncludeInRelease
671 <latexrelease>
672 <latexrelease>\IncludeInRelease{2020/10/01}%
673 <latexrelease> {\normalfont}{Add hook to \normalfont}%
674 <latexrelease>
675 <latexrelease>\DeclareRobustCommand\normalfont{%
676 <latexrelease> \fontencoding\encodingdefault
677 <latexrelease> \edef\f@family{\familydefault}%
678 <latexrelease> \edef\f@series{\seriesdefault}%
679 <latexrelease> \edef\f@shape{\shapedefault}%
680 <latexrelease> \UseHook{normalfont}%
681 <latexrelease> \@defaultfamilyhook % hookname from 2020/02 will vanish
682 <latexrelease> \selectfont}
683 <latexrelease>
684 <latexrelease>\let\reset@font\normalfont
685 <latexrelease>
686 <latexrelease>\EndIncludeInRelease
687 <latexrelease>
688 <latexrelease>\IncludeInRelease{2020/02/02}%
689 <latexrelease> {\normalfont}{Add hook to \normalfont}%
690 <latexrelease>
691 <latexrelease>\DeclareRobustCommand\normalfont{%
692 <latexrelease> \fontencoding\encodingdefault
693 <latexrelease> \edef\f@family{\familydefault}%
694 <latexrelease> \edef\f@series{\seriesdefault}%
695 <latexrelease> \edef\f@shape{\shapedefault}%
696 <latexrelease> \@defaultfamilyhook
697 <latexrelease> \selectfont}
698 <latexrelease>
699 <latexrelease>\let\reset@font\normalfont
700 <latexrelease>
701 <latexrelease>\let@\defaultfamilyhook@empty
702 <latexrelease>
703 <latexrelease>\EndIncludeInRelease
704 <latexrelease>
705 <latexrelease>\IncludeInRelease{0000/00/00}%
706 <latexrelease> {\normalfont}{Add hook to \normalfont}%
707 <latexrelease>
708 <latexrelease>\DeclareRobustCommand\normalfont
709 <latexrelease> {\usefont\encodingdefault
710 <latexrelease> \familydefault
711 <latexrelease> \seriesdefault
712 <latexrelease> \shapedefault
```

```

713 \relax}
714 \let\reset@font\normalfont
715 \let\@defaultfamilyhook\undefined
716 \let\EndIncludeInRelease
717 \EndIncludeInRelease
718 \EndIncludeInRelease
719 {*2ekernel}

```

We left out the special L<sup>A</sup>T<sub>E</sub>X fonts which are not automatically included in the base version of the font selection since these fonts contain only a few characters which are also included in the AMS fonts so anybody who is using these fonts doesn't need them. But for compatibility reasons we will define these symbols.

```

720 \def\not@base#1{\@latex@error
721 {Command \noexpand#1 not provided in base LATEX2e}%
722 {Load the latexsym or the amsfonts package to
723 define this symbol}}
724 \def\mho{\not@base\mho}
725 \def\Join{\not@base\Join}
726 \def\Box{\not@base\Box}
727 \def\Diamond{\not@base\Diamond}
728 \def\leadsto{\not@base\leadsto}
729 \def\sqsubset{\not@base\sqsubset}
730 \def\sqsupset{\not@base\sqsupset}
731 \def\lhd{\not@base\lhd}
732 \def\unlhd{\not@base\unlhd}
733 \def\rhd{\not@base\rhd}
734 \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.

```

735 \DeclareErrorFont{OT1}{cmr}{m}{n}{10} %% don't modify this setting
736 %% overwrite it in fontdef.cfg
737 %% 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`.

```

738 \fontfamily{cmr}

```

Previously the default values for series and shape were set by calling `\fontseries` and `\fontshape`, but their action is now delayed until `\selectfont` which isn't called inside the format (to avoid unnecessarily loading a font that may never get used). We therefore have to set `\f@series` and `\f@shape` directly instead.

```

739 \def\f@series{m} % \fontseries{m}
740 \def\f@shape{n} % \fontshape{n}
741 \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:

```

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

```

We now load the customizable parts of NFSS.

```
743 \InputIfFileExists{fonttext.cfg}
744 {\typeout{=====
745 ^^^J%
746 Local config file fonttext.cfg used^ ^^J%
747 ^^^J%
748 =====}%
749 \def\@addtolist##1{\xdef\@filelist{\@filelist,##1}}%
750 }
751 {\input{fonttext.ltx}}
752 \let\@addtolist\@gobble
Ditto for math although I don't think that we will get a lot of customisation :-)
753 \InputIfFileExists{fontmath.cfg}
754 {\typeout{=====
755 ^^^J%
756 Local config file fontmath.cfg used^ ^^J%
757 ^^^J%
758 =====}%
759 \def\@addtolist##1{\xdef\@filelist{\@filelist,##1}}%
760 }
761 {\input{fontmath.ltx}}
762 \let\@addtolist\@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`.

```
763 \InputIfFileExists{preload.cfg}
764 {\typeout{=====
765 ^^^J%
766 Local config file preload.cfg used^ ^^J%
767 ^^^J%
768 =====}%
769 \def\@addtolist##1{\xdef\@filelist{\@filelist,##1}}%
770 }
771 {\input{preload.ltx}}
772 \let\@addtolist\@gobble
```

\seriesdefault 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`.

```
773 \expandafter\def\expandafter\seriesdefault\expandafter{\seriesdefault\empty}
774 \let\seriesdefault@kernel\seriesdefault
```

(End of definition for `\seriesdefault` and `\seriesdefault@kernel`.)

\@acci We also save the values of some accents in `\@acci`, `\@accii` and `\@acciii` so they can  
\@accii be restored by a `minipage` inside a `tabbing` environment.

```
775 \let\@acci\` \let\@accii\` \let\@acciii\=
```

(End of definition for `\@acci`, `\@accii`, and `\@acciii`.)

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

*(End of definition for \cal and \mit.)*

776 ⟨/2ekernel⟩

# File 30

## fontdef.dtx

### 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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  finds a file of this name it will use it, otherwise it uses the standard file which is `fontdef.ltx`.

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

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

**Warning:** please note that we don't support customised L<sup>A</sup>T<sub>E</sub>X versions. Thus, before sending in a bug report please try your test file with a L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X installations behave in the same way.

|     |                                                      |
|-----|------------------------------------------------------|
| T1  | Cork T <sub>E</sub> X text encoding                  |
| OT1 | old T <sub>E</sub> X text encoding                   |
| U   | unknown encoding                                     |
| OML | old T <sub>E</sub> X math letters encoding           |
| OMS | old T <sub>E</sub> X math symbols encoding           |
| OMX | old T <sub>E</sub> X math extension symbols encoding |
| TU  | Unicode                                              |

Notice that some of these encodings are ‘old’ in the sense that we hope that they will be superseded soon by encoding standards defined by the  $\text{\TeX}$  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  $\text{\TeX}$  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  $\text{\LaTeX}$  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 <code>fonttext.ltx</code>     |
| math    | produce the file <code>fontmath.ltx</code>     |
| cfgtext | produce a dummy <code>fonttext.cfg</code> file |
| cfgmath | produce a dummy <code>fontmath.cfg</code> file |

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

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

### 4 A driver for this document

The next bit of code contains the documentation driver file for  $\text{\TeX}$ , i.e. the file that will produce the documentation you are currently reading. It will be extracted from this file by the `DOCSTRIP` program.

```
1 <*driver>
2 \documentclass{ltxdoc}
3 \GetFileInfo{fontdef.dtx}
4 \begin{document}
5 \DocInput{fontdef.dtx}
6 \end{document}
7
```

### 5 The `fonttext.ltx` file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
8 <*text>
9 \typeout{== Don't modify this file, use a .cfg file instead ==^J}
```

## 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 `loutenc.dtx`.

By convention, text encoding specific declarations, including the `\DeclareFontEncoding` declaration, are kept in separate file of the form `<enc>enc.def`, e.g. `ot1enc.def`. This allows other applications to make use of the declarations as well.

Similar to the default encoding, the loading of the encoding files for the two major text encodings shouldn't be changed. In particular, the `inputenc` package depends on this.

```
10 \input {omlenc.def}
11 \input {omsenc.def}

12 \input {ot1enc.def}
13 \input {t1enc.def}
14 \input{ts1enc.def}

15 \ifx\Umathcode\@undefined
```

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`).

```
16 \fontencoding{OT1}
```

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.

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

```
18 \def\rmsubstdefault{cmr}
19 \def\sfsubstdefault{cmss}
20 \def\ttsubstdefault{cmtt}
```

```
21 \LoadFontDefinitionFile{TS1}{cmr}
```

```
22 \else
```

Unicode.  
23 \input {tuenc.def}

```
24 \fontencoding{TU}
```

The initial `fontenc` package load list if a Unicode engine is used:

```
25 \def\@fontenc@load@list{\@elt{TU}}
```

```
26 \DeclareFontSubstitution{TU}{lmr}{m}{n}
```

```
27 \LoadFontDefinitionFile{TU}{lmr}
```

```
28 \LoadFontDefinitionFile{TU}{lmss}
```

```
29 \LoadFontDefinitionFile{TU}{lmtt}
```

```
30 \def\rmsubstdefault{lmr}
```

```
31 \def\sfsubstdefault{lmss}
```

```
32 \def\ttsubstdefault{lmmtt}
```

```
33 \LoadFontDefinitionFile{TS1}{lmr}
```

```
34 \DeclareFontSubstitution{TU}{lmr}{m}{n}
```

End of Unicode branch.

```
35 \fi
```

If different encodings for text fonts are in use one could put the common setup into `\DeclareFontEncodingDefaults`. There is now a better mechanism so using this interface is discouraged!

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

Then we define the default substitution for every encoding. This release of L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  assumes that the ec fonts are available. It is possible to change this to point to some other font family (e.g., Times with the appropriate encoding if it is available) without making documents non-portable. However, in such a case documents will produce different page breaks at other sites. The substitution defaults can all be changed without losing portability as long as there are font shape definitions for the selected substitutions.

```
37 \DeclareFontSubstitution{T1}{cmr}{m}{n}
```

```
38 \DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration, L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  will try to verify that the given substitution information makes sense, i.e. that it is impossible to go into an endless loop if font substitution happens. This is done at the moment the `\begin{document}` is encountered. L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  will then check that for every encoding the substitution defaults form a valid font shape group, which means that it will check if there is a `\DeclareFontShape` declaration for this combination. We will therefore load the corresponding .fd files now. If we don't do this they would be loaded at verification time (i.e. at `\begin{document}`) which would delay processing unnecessarily.

**Warning:** Please note that this means that you have to regenerate the format whenever you change any of these .fd files since L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  will not read .fd files if it already knows about the encoding/family combination.

The `\nfss@catcodes` ensures that white space is ignored in any definitions made in the fd files.

```
39 \begingroup
40 \nfss@catcodes
41 \input {t1cmr.fd}
42 \input {ot1cmr.fd}
43 \endgroup
```

We also load some other font definition files which are normally needed in a document. This is only done for processing speed and you can comment the next two lines out to save some memory. If necessary these files are then loaded when your document is processed. (Loading .fd files is a less drastic step compared to preloading fonts because the number of fonts is limited 255 at (nearly) every T<sub>E</sub>X installation, while the amount of main memory is not a limiting factor at most installations.)

```
44 \begingroup
45 \nfss@catcodes
46 \input {t1cmss.fd}
47 \input {t1cmtt.fd}
48 \input {ot1cmss.fd}
49 \input {ot1cmtt.fd}
50 \endgroup
```

Even though Unicode engines default to `\load{ts1cmr}` as this may be used for fallback for TS1 encoding.

```

51 \ifx\Umathcode\@undefined\else
52 \begingroup
53 \nfss@catcodes
54 \input {ts1cmr.fd}
55 \endgroup
56 \fi

```

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.

```

57 \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 The following three definitions set up the meaning for \rmfamily, \sffamily, and
\rmdefault \ttfamily.
\sfdefault
58 \ifx\Umathcode\@undefined
59 \newcommand\encodingdefault{OT1}
60 \newcommand\rmdefault{cmr}
61 \newcommand\sfdefault{cmss}
62 \newcommand\ttdefault{cmtt}
63 \else
64 \newcommand\encodingdefault{TU}
65 \newcommand\rmdefault{lmr}
66 \fontfamily{\rmdefault}
67 \newcommand\sfdefault{lmss}
68 \newcommand\ttdefault{lmtt}
69 \fi
70
71 <tex>\IncludeInRelease{2017/01/01}%
72 <tex>\encodingdefault{TU encoding default}%
73 <tex>\ifx\Umathcode\@undefined
74 <tex>\renewcommand\encodingdefault{OT1}
75 <tex>\fontencoding{\encodingdefault}
76 <tex>\renewcommand\rmdefault{cmr}
77 <tex>\fontfamily{\rmdefault}
78 <tex>\renewcommand\sfdefault{cmss}
79 <tex>\renewcommand\ttdefault{cmtt}
80 <tex>\else
81 <tex>\renewcommand\encodingdefault{TU}

```

```

82 〈latexrelease〉%done in everyjob\fontencoding{\encodingdefault}
83 〈latexrelease〉\renewcommand\rmdefault{lmr}
84 〈latexrelease〉\fontfamily{\rmdefault}
85 〈latexrelease〉\renewcommand\sffont{lmss}
86 〈latexrelease〉\renewcommand\ttfont{lmmtt}
87 〈latexrelease〉\fi
88 〈latexrelease〉\EndIncludeInRelease
89 〈latexrelease〉\IncludeInRelease{0000/00/00}%
90 〈latexrelease〉\encodingdefault{\encodingdefault}{TU encoding default}%
91 〈latexrelease〉\fontencoding{OT1}
92 〈latexrelease〉\renewcommand\encodingdefault{OT1}
93 〈latexrelease〉\fontencoding{\encodingdefault}
94 〈latexrelease〉\renewcommand\rmdefault{cmr}
95 〈latexrelease〉\fontfamily{\rmdefault}
96 〈latexrelease〉\renewcommand\sffont{cmss}
97 〈latexrelease〉\renewcommand\ttfont{cmtt}
98 〈latexrelease〉\EndIncludeInRelease
99 〈*text〉

```

(End of definition for `\encodingdefault` and others.)

`\bfdefault` Series changing commands are influenced by the following hooks.  
`\mddefault` 100 `\newcommand\bfdefault{b}` % overwritten below (for rollback)  
101 `\newcommand\mddefault{m}` % overwritten below (for rollback)

(End of definition for `\bfdefault` and `\mddefault`.)

`\itdefault` Shape changing commands use the following hooks.  
`\sldefault` 102 `\newcommand\itdefault{it}`  
`\scdefault` 103 `\newcommand\sldefault{sl}`  
`\updefault` 104 `\newcommand\scdefault{sc}`  
105 `\newcommand\updefault{up}` % overwritten below (for rollback)

(End of definition for `\itdefault` and others.)

```

106 〈/text〉
107 〈*text | latexrelease〉
108 〈latexrelease〉\IncludeInRelease{2020/02/02}%
109 〈latexrelease〉\updefault{\updefault}{font defaults change}%
110 \renewcommand\updefault{up}

```

We append `\empty` to the series value so that we can detect if it got changed via `\def` or `\renewcommand` later.

```

111 \renewcommand\bfdefault{b\empty}
112 \renewcommand\mddefault{m\empty}
113 \let\bfdefault@previous\bfdefault
114 \let\mddefault@previous\mddefault
115 〈/text | latexrelease〉
116 〈latexrelease〉\EndIncludeInRelease
117 〈latexrelease〉\IncludeInRelease{0000/00/00}%
118 〈latexrelease〉\updefault{\updefault}{font defaults change}%
119 〈latexrelease〉
120 〈latexrelease〉\renewcommand\updefault{n}
121 〈latexrelease〉\renewcommand\bfdefault{bx}
122 〈latexrelease〉

```

```

123 ⟨latexrelease⟩\let\bfdefault@previous\undefined
124 ⟨latexrelease⟩\let\mddefault@previous\undefined
125 ⟨latexrelease⟩\EndIncludeInRelease
126 ⟨*text⟩

```

`\familydefault` Finally we have the hooks that describe the behaviour of the `\normalfont` command.  
`\seriesdefault` To stay portable, the definition of `\encodingdefault` should *not* be changed and should  
`\shapedefault` match the setting above for `\fontencoding`. All other values can be set according to  
your taste.

```

127 \newcommand\familydefault{\rmdefault}
128 \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.

```
129 \newcommand\shapedefault{n}
```

(*End of definition for `\familydefault`, `\seriesdefault`, and `\shapedefault`.*)

This finishes the low-level setup in `fonttext.ltx`.

```
130 ⟨/text⟩
```

## 6 The `fontmath.ltx` file

The identification is done earlier on with a `\ProvidesFile` declaration.

```

131 ⟨*math⟩
132 \typeout{== Don't modify this file, use a .cfg file instead ==^J}

```

### 6.1 The font encodings used

```

133 \DeclareFontEncoding{OML}{}{}
134 \DeclareFontEncoding{OMS}{}{}
135 \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-L<sup>A</sup>T<sub>E</sub>X. 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.

```
136 \DeclareFontEncoding{U}{}{\noaccents@}
```

The encodings for math are next:

```

137 \DeclareFontSubstitution{OML}{cmm}{m}{it}
138 \DeclareFontSubstitution{OMS}{cmsy}{m}{n}
139 \DeclareFontSubstitution{OMX}{cmex}{m}{n}
140 \DeclareFontSubstitution{U}{cmr}{m}{n}

141 \begingroup
142 \nfss@catcodes
143 \input {omlcmm.fd}
144 \input {omscmsy.fd}
145 \input {omxcmex.fd}
146 \input {ucmr.fd}
147 \endgroup

```

### 6.1.1 Symbolfont and Alphabet declarations

We now define the basic symbol fonts used by L<sup>A</sup>T<sub>E</sub>X. These four symbol fonts must be defined by this file.

It is possible to make the symbol fonts point to other external fonts without losing the ability to process documents written at other sites, as long as one defines the same symbol font names with the same encodings, e.g. operators with OT1 etc. If other encodings are used documents become non-portable. Such a change should therefore be done in a package file.

```
148 \DeclareSymbolFont{operators} {OT1}{cmr}{m}{n}
149 \DeclareSymbolFont{letters} {OML}{cmm}{m}{it}
150 \DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}
151 \DeclareSymbolFont{largetsymbol}{OMX}{cmex}{m}{n}
152 \SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
153 \SetSymbolFont{letters} {bold}{OML}{cmm}{b}{it}
154 \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.

```
155 \DeclareSymbolFontAlphabet{\mathrm}{operators}
156 \DeclareSymbolFontAlphabet{\mathnormal}{letters}
157 \DeclareSymbolFontAlphabet{\mathcal}{symbols}
158 \DeclareMathAlphabet{\mathbf}{OT1}{cmr}{bx}{n}
159 \DeclareMathAlphabet{\mathsf}{OT1}{cmss}{m}{n}
160 \DeclareMathAlphabet{\mathit}{OT1}{cmr}{m}{it}
161 \DeclareMathAlphabet{\mathtt}{OT1}{cmss}{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.

```
162 \SetMathAlphabet{\mathsf}{bold}{OT1}{cmss}{bx}{n}
163 \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).

```
164 \DeclareMathSizes{5}{5}{5}{5}
165 \DeclareMathSizes{6}{6}{5}{5}
166 \DeclareMathSizes{7}{7}{5}{5}
167 \DeclareMathSizes{8}{8}{6}{5}
168 \DeclareMathSizes{9}{9}{6}{5}
169 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
170 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
171 \DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
172 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
```

```

173 \DeclareMathSizes{\@xviipt}{\@xviipt}{\@xiipt}{\@xipt}
174 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
175 \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

```

176 \DeclareMathSymbol{a}{\mathalpha}{letters}{`a}
177 \DeclareMathSymbol{b}{\mathalpha}{letters}{`b}
178 \DeclareMathSymbol{c}{\mathalpha}{letters}{`c}
179 \DeclareMathSymbol{d}{\mathalpha}{letters}{`d}
180 \DeclareMathSymbol{e}{\mathalpha}{letters}{`e}
181 \DeclareMathSymbol{f}{\mathalpha}{letters}{`f}
182 \DeclareMathSymbol{g}{\mathalpha}{letters}{`g}
183 \DeclareMathSymbol{h}{\mathalpha}{letters}{`h}
184 \DeclareMathSymbol{i}{\mathalpha}{letters}{`i}
185 \DeclareMathSymbol{j}{\mathalpha}{letters}{`j}
186 \DeclareMathSymbol{k}{\mathalpha}{letters}{`k}
187 \DeclareMathSymbol{l}{\mathalpha}{letters}{`l}
188 \DeclareMathSymbol{m}{\mathalpha}{letters}{`m}
189 \DeclareMathSymbol{n}{\mathalpha}{letters}{`n}
190 \DeclareMathSymbol{o}{\mathalpha}{letters}{`o}
191 \DeclareMathSymbol{p}{\mathalpha}{letters}{`p}
192 \DeclareMathSymbol{q}{\mathalpha}{letters}{`q}
193 \DeclareMathSymbol{r}{\mathalpha}{letters}{`r}
194 \DeclareMathSymbol{s}{\mathalpha}{letters}{`s}
195 \DeclareMathSymbol{t}{\mathalpha}{letters}{`t}
196 \DeclareMathSymbol{u}{\mathalpha}{letters}{`u}
197 \DeclareMathSymbol{v}{\mathalpha}{letters}{`v}
198 \DeclareMathSymbol{w}{\mathalpha}{letters}{`w}
199 \DeclareMathSymbol{x}{\mathalpha}{letters}{`x}
200 \DeclareMathSymbol{y}{\mathalpha}{letters}{`y}
201 \DeclareMathSymbol{z}{\mathalpha}{letters}{`z}

202 \DeclareMathSymbol{A}{\mathalpha}{letters}{`A}
203 \DeclareMathSymbol{B}{\mathalpha}{letters}{`B}
204 \DeclareMathSymbol{C}{\mathalpha}{letters}{`C}
205 \DeclareMathSymbol{D}{\mathalpha}{letters}{`D}
206 \DeclareMathSymbol{E}{\mathalpha}{letters}{`E}
207 \DeclareMathSymbol{F}{\mathalpha}{letters}{`F}
208 \DeclareMathSymbol{G}{\mathalpha}{letters}{`G}
209 \DeclareMathSymbol{H}{\mathalpha}{letters}{`H}
210 \DeclareMathSymbol{I}{\mathalpha}{letters}{`I}
211 \DeclareMathSymbol{J}{\mathalpha}{letters}{`J}
212 \DeclareMathSymbol{K}{\mathalpha}{letters}{`K}
213 \DeclareMathSymbol{L}{\mathalpha}{letters}{`L}
214 \DeclareMathSymbol{M}{\mathalpha}{letters}{`M}
215 \DeclareMathSymbol{N}{\mathalpha}{letters}{`N}

```

```

216 \DeclareMathSymbol{O}{\mathalpha}{letters}{`0}
217 \DeclareMathSymbol{P}{\mathalpha}{letters}{`P}
218 \DeclareMathSymbol{Q}{\mathalpha}{letters}{`Q}
219 \DeclareMathSymbol{R}{\mathalpha}{letters}{`R}
220 \DeclareMathSymbol{S}{\mathalpha}{letters}{`S}
221 \DeclareMathSymbol{T}{\mathalpha}{letters}{`T}
222 \DeclareMathSymbol{U}{\mathalpha}{letters}{`U}
223 \DeclareMathSymbol{V}{\mathalpha}{letters}{`V}
224 \DeclareMathSymbol{W}{\mathalpha}{letters}{`W}
225 \DeclareMathSymbol{X}{\mathalpha}{letters}{`X}
226 \DeclareMathSymbol{Y}{\mathalpha}{letters}{`Y}
227 \DeclareMathSymbol{Z}{\mathalpha}{letters}{`Z}

```

### 6.3.2 The digits

```

228 \DeclareMathSymbol{0}{\mathalpha}{operators}{`0}
229 \DeclareMathSymbol{1}{\mathalpha}{operators}{`1}
230 \DeclareMathSymbol{2}{\mathalpha}{operators}{`2}
231 \DeclareMathSymbol{3}{\mathalpha}{operators}{`3}
232 \DeclareMathSymbol{4}{\mathalpha}{operators}{`4}
233 \DeclareMathSymbol{5}{\mathalpha}{operators}{`5}
234 \DeclareMathSymbol{6}{\mathalpha}{operators}{`6}
235 \DeclareMathSymbol{7}{\mathalpha}{operators}{`7}
236 \DeclareMathSymbol{8}{\mathalpha}{operators}{`8}
237 \DeclareMathSymbol{9}{\mathalpha}{operators}{`9}

```

### 6.3.3 Punctuation, brace, etc. keys

```

238 \DeclareMathSymbol{!}{\mathclose}{operators}{`21}
239 \DeclareMathSymbol{*}{\mathbin}{symbols}{`03} % \ast
240 \DeclareMathSymbol{+}{\mathbin}{operators}{`2B}
241 \DeclareMathSymbol{,}{\mathpunct}{letters}{`3B}
242 \DeclareMathSymbol{-}{\mathbin}{symbols}{`00}
243 \DeclareMathSymbol{.}{\mathord}{letters}{`3A}
244 \DeclareMathSymbol{:}{\mathrel}{operators}{`3A}
245 \DeclareMathSymbol{;}{\mathpunct}{operators}{`3B}
246 \DeclareMathSymbol{=}{\mathrel}{operators}{`3D}
247 \DeclareMathSymbol{?}{\mathclose}{operators}{`3F}

```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```

248 \% \DeclareMathSymbol{()}{\mathopen}{operators}{`28}
249 \% \DeclareMathSymbol{}{\mathclose}{operators}{`29}
250 \% \DeclareMathSymbol{/}{\mathord}{letters}{`3D}
251 \% \DeclareMathSymbol{[]}{\mathopen}{operators}{`5B}
252 \% \DeclareMathSymbol{}{\mathclose}{operators}{`5D}
253 \% \DeclareMathSymbol{|}{\mathord}{symbols}{`6A}
254 \% \DeclareMathSymbol{<}{\mathrel}{letters}{`3C}
255 \% \DeclareMathSymbol{>}{\mathrel}{letters}{`3E}

```

Should all of the following being activated by default? Probably not.

```

256 \% \DeclareMathSymbol{'{}{\mathopen}{symbols}{`66}
257 \% \DeclareMathSymbol{'{}{\mathclose}{symbols}{`67}
258 \% \DeclareMathSymbol{'\\}{\mathord}{symbols}{`6E} % \backslash
259 \mathcode`\ = "8000 % \space
260 \mathcode`'= "8000 % ^\prime
261 \mathcode`_ = "8000 % \

```

### 6.3.4 Delimitercodes for characters

[to be completed]

Finally, InTiEX sets all `\delcode` values to -1, except `\delcode`.=0`

```

262 \DeclareMathDelimiter{()}{\mathopen}{operators}{28}{largesymbols}{00}
263 \DeclareMathDelimiter{}{\mathclose}{operators}{29}{largesymbols}{01}
264 \DeclareMathDelimiter{[]}{\mathopen}{operators}{5B}{largesymbols}{02}
265 \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.

```

266 \DeclareMathDelimiter{<}{\mathopen}{symbols}{68}{largesymbols}{0A}
267 \DeclareMathDelimiter{>}{\mathclose}{symbols}{69}{largesymbols}{0B}
268 \DeclareMathSymbol{<}{\mathrel}{letters}{3C}
269 \DeclareMathSymbol{>}{\mathrel}{letters}{3E}
```

And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```

270 \DeclareMathDelimiter{/}{\mathord}{operators}{2F}{largesymbols}{0E}
271 \DeclareMathSymbol{/}{\mathord}{letters}{3D}
272 \DeclareMathDelimiter{|}{\mathord}{symbols}{6A}{largesymbols}{0C}
273 \expandafter\DeclareMathDelimiter@backslashchar
274 \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

```

275 \DeclareMathSymbol{\alpha}{\mathord}{letters}{0B}
276 \DeclareMathSymbol{\beta}{\mathord}{letters}{0C}
277 \DeclareMathSymbol{\gamma}{\mathord}{letters}{0D}
278 \DeclareMathSymbol{\delta}{\mathord}{letters}{0E}
279 \DeclareMathSymbol{\epsilon}{\mathord}{letters}{0F}
280 \DeclareMathSymbol{\zeta}{\mathord}{letters}{10}
281 \DeclareMathSymbol{\eta}{\mathord}{letters}{11}
282 \DeclareMathSymbol{\theta}{\mathord}{letters}{12}
283 \DeclareMathSymbol{\iota}{\mathord}{letters}{13}
284 \DeclareMathSymbol{\kappa}{\mathord}{letters}{14}
285 \DeclareMathSymbol{\lambda}{\mathord}{letters}{15}
286 \DeclareMathSymbol{\mu}{\mathord}{letters}{16}
287 \DeclareMathSymbol{\nu}{\mathord}{letters}{17}
288 \DeclareMathSymbol{\xi}{\mathord}{letters}{18}
289 \DeclareMathSymbol{\pi}{\mathord}{letters}{19}
290 \DeclareMathSymbol{\rho}{\mathord}{letters}{1A}
291 \DeclareMathSymbol{\sigma}{\mathord}{letters}{1B}
292 \DeclareMathSymbol{\tau}{\mathord}{letters}{1C}
293 \DeclareMathSymbol{\upsilon}{\mathord}{letters}{1D}
294 \DeclareMathSymbol{\phi}{\mathord}{letters}{1E}
295 \DeclareMathSymbol{\chi}{\mathord}{letters}{1F}
296 \DeclareMathSymbol{\psi}{\mathord}{letters}{20}
```

```

297 \DeclareMathSymbol{\omega}{\mathord}{letters}{21}
298 \DeclareMathSymbol{\varepsilon}{\mathord}{letters}{22}
299 \DeclareMathSymbol{\vartheta}{\mathord}{letters}{23}
300 \DeclareMathSymbol{\varpi}{\mathord}{letters}{24}
301 \DeclareMathSymbol{\varrho}{\mathord}{letters}{25}
302 \DeclareMathSymbol{\varsigma}{\mathord}{letters}{26}
303 \DeclareMathSymbol{\varphi}{\mathord}{letters}{27}
304 \DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{00}
305 \DeclareMathSymbol{\Delta}{\mathalpha}{operators}{01}
306 \DeclareMathSymbol{\Theta}{\mathalpha}{operators}{02}
307 \DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{03}
308 \DeclareMathSymbol{\Xi}{\mathalpha}{operators}{04}
309 \DeclareMathSymbol{\Pi}{\mathalpha}{operators}{05}
310 \DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{06}
311 \DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{07}
312 \DeclareMathSymbol{\Phi}{\mathalpha}{operators}{08}
313 \DeclareMathSymbol{\Psi}{\mathalpha}{operators}{09}
314 \DeclareMathSymbol{\Omega}{\mathalpha}{operators}{0A}

```

#### 6.4.2 Ordinary symbols

```

315 \DeclareMathSymbol{\aleph}{\mathord}{symbols}{40}
316 \DeclareMathSymbol{\imath}{\mathord}{letters}{7B}
317 \DeclareMathSymbol{\jmath}{\mathord}{letters}{7C}
318 \DeclareMathSymbol{\ell}{\mathord}{letters}{60}
319 \DeclareMathSymbol{\wp}{\mathord}{letters}{7D}
320 \DeclareMathSymbol{\Re}{\mathord}{symbols}{3C}
321 \DeclareMathSymbol{\Im}{\mathord}{symbols}{3D}
322 \DeclareMathSymbol{\partial}{\mathord}{letters}{40}
323 \DeclareMathSymbol{\infty}{\mathord}{symbols}{31}
324 \DeclareMathSymbol{\prime}{\mathord}{symbols}{30}
325 \DeclareMathSymbol{\emptyset}{\mathord}{symbols}{3B}
326 \DeclareMathSymbol{\nabla}{\mathord}{symbols}{72}
327 \DeclareMathSymbol{\top}{\mathord}{symbols}{3E}
328 \DeclareMathSymbol{\bot}{\mathord}{symbols}{3F}
329 \DeclareMathSymbol{\triangle}{\mathord}{symbols}{34}
330 \DeclareMathSymbol{\forall}{\mathord}{symbols}{38}
331 \DeclareMathSymbol{\exists}{\mathord}{symbols}{39}
332 \DeclareMathSymbol{\neg}{\mathord}{symbols}{3A}

```

Alias:

```

333 % \let\lnot=\neg
334 \DeclareMathSymbol{\lnot}{\mathord}{symbols}{3A}
335 \DeclareMathSymbol{\flat}{\mathord}{letters}{5B}
336 \DeclareMathSymbol{\natural}{\mathord}{letters}{5C}
337 \DeclareMathSymbol{\sharp}{\mathord}{letters}{5D}
338 \DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{7C}
339 \DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{7D}
340 \DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{7E}
341 \DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{7F}
342 \ DeclareRobustCommand{\hbar}{\mathchar'26\mkern-9mu h}
343 \ DeclareRobustCommand{\surd}{\mathchar"1270}
344 \ DeclareRobustCommand{\angle}{\vbox{\ialign{$\m@th\scriptstyle##$\crcr
345 \not\mathrel{\mkern14mu}\crcr
346 \noalign{\nointerlineskip}

```

```
347 \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crcr\}}}\}
```

### 6.4.3 Large Operators

```
348 \DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{60}
349 \DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{57}
350 \DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{56}
351 \DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{55}
352 \DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{54}
353 \DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{53}
354 \DeclareMathSymbol{\intop}{\mathop}{largesymbols}{52}
355 \ DeclareRobustCommand\int{\intop\nolimits}
356 \DeclareMathSymbol{\prod}{\mathop}{largesymbols}{51}
357 \DeclareMathSymbol{\sum}{\mathop}{largesymbols}{50}
358 \DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{4E}
359 \DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{4C}
360 \DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{4A}
361 \DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{48}
362 \ DeclareRobustCommand\oint{\ointop\nolimits}
363 \DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{46}
364 \DeclareMathSymbol{\smallint}{\mathop}{symbols}{73}
```

### 6.4.4 Binary symbols

```
365 \DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{2F}
366 \DeclareMathSymbol{\triangleright}{\mathbin}{letters}{2E}
367 \DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{34}
368 \DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{35}
```

Alias:

```
369 % \let \varbigtriangledown \bigtriangledown
370 % \let \varbigtriangleup \bigtriangleup
371 \DeclareMathSymbol{\varbigtriangleup}{\mathbin}{symbols}{34}
372 \DeclareMathSymbol{\varbigtriangledown}{\mathbin}{symbols}{35}
```

These last two synonyms are needed because the stmaryrd package redefines them as Operators.

```
373 \DeclareMathSymbol{\wedge}{\mathbin}{symbols}{5E}
374 \DeclareMathSymbol{\vee}{\mathbin}{symbols}{5F}
```

Alias:

```
375 % \let\land=\wedge
376 % \let\lor=\vee
377 \DeclareMathSymbol{\land}{\mathbin}{symbols}{5E}
378 \DeclareMathSymbol{\lor}{\mathbin}{symbols}{5F}
379 \DeclareMathSymbol{\cap}{\mathbin}{symbols}{5C}
380 \DeclareMathSymbol{\cup}{\mathbin}{symbols}{5B}
381 \DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{7A}
382 \DeclareMathSymbol{\dagger}{\mathbin}{symbols}{79}
383 \DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{75}
384 \DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{74}
385 \DeclareMathSymbol{\uplus}{\mathbin}{symbols}{5D}
386 \DeclareMathSymbol{\amalg}{\mathbin}{symbols}{71}
387 \DeclareMathSymbol{\diamond}{\mathbin}{symbols}{05}
388 \DeclareMathSymbol{\bullet}{\mathbin}{symbols}{0F}
389 \DeclareMathSymbol{\wr}{\mathbin}{symbols}{6F}
390 \DeclareMathSymbol{\div}{\mathbin}{symbols}{04}
```

```

391 \DeclareMathSymbol{\odot}{\mathbin}{symbols}{0C}
392 \DeclareMathSymbol{\oslash}{\mathbin}{symbols}{0B}
393 \DeclareMathSymbol{\otimes}{\mathbin}{symbols}{0A}
394 \DeclareMathSymbol{\ominus}{\mathbin}{symbols}{09}
395 \DeclareMathSymbol{\oplus}{\mathbin}{symbols}{08}
396 \DeclareMathSymbol{\mp}{\mathbin}{symbols}{07}
397 \DeclareMathSymbol{\pm}{\mathbin}{symbols}{06}
398 \DeclareMathSymbol{\circ}{\mathbin}{symbols}{0E}
399 \DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{0D}
400 \DeclareMathSymbol{\setminus}{\mathbin}{symbols}{6E}
401 \DeclareMathSymbol{\cdotp}{\mathbin}{symbols}{01}
402 \DeclareMathSymbol{\ast}{\mathbin}{symbols}{03}
403 \DeclareMathSymbol{\times}{\mathbin}{symbols}{02}
404 \DeclareMathSymbol{\star}{\mathbin}{letters}{3F}

```

#### 6.4.5 Relations

```

405 \DeclareMathSymbol{\propto}{\mathrel}{symbols}{2F}
406 \DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{76}
407 \DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{77}
408 \DeclareMathSymbol{\parallel}{\mathrel}{symbols}{6B}
409 \DeclareMathSymbol{\mid}{\mathrel}{symbols}{6A}
410 \DeclareMathSymbol{\dashv}{\mathrel}{symbols}{61}
411 \DeclareMathSymbol{\vdash}{\mathrel}{symbols}{60}
412 \DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{25}
413 \DeclareMathSymbol{\searrow}{\mathrel}{symbols}{26}
414 \DeclareMathSymbol{\nwarrow}{\mathrel}{symbols}{2D}
415 \DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{2E}
416 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{2C}
417 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{28}
418 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{29}
419 \DeclareRobustCommand{\neq}{\not=}

```

As `\neq` is robust we should not use `\let` to define `\ne` as then it would change if `\neq` changes.

```
420 \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).

```

421 \DeclareMathSymbol{\leq}{\mathrel}{symbols}{14}
422 \DeclareMathSymbol{\geq}{\mathrel}{symbols}{15}

```

Alias:

```

423 % \let\le=\leq
424 % \let\ge=\geq
425 \DeclareMathSymbol{\le}{\mathrel}{symbols}{14}
426 \DeclareMathSymbol{\ge}{\mathrel}{symbols}{15}
427 \DeclareMathSymbol{\succ}{\mathrel}{symbols}{1F}
428 \DeclareMathSymbol{\prec}{\mathrel}{symbols}{1E}
429 \DeclareMathSymbol{\approx}{\mathrel}{symbols}{19}
430 \DeclareMathSymbol{\succeq}{\mathrel}{symbols}{17}
431 \DeclareMathSymbol{\preceq}{\mathrel}{symbols}{16}
432 \DeclareMathSymbol{\supset}{\mathrel}{symbols}{1B}
433 \DeclareMathSymbol{\subset}{\mathrel}{symbols}{1A}
434 \DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{13}
435 \DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{12}

```

```

436 \DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
437 \DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}

Alias:
438 % \let\owns=\ni
439 \DeclareMathSymbol{\owns}{\mathrel}{symbols}{"33}
440 \DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
441 \DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
442 \DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
443 \DeclareMathSymbol{\leftrightarrow}{\mathrel}{symbols}{"24}
444 \DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
445 \DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}

Alias:
446 % \let\gets=\leftarrow
447 % \let\to=\rightarrow
448 \DeclareMathSymbol{\gets}{\mathrel}{symbols}{"20}
449 \DeclareMathSymbol{\to}{\mathrel}{symbols}{"21}

450 \DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
451 \ DeclareRobustCommand{\mapsto{\mapstochar\rightarrow}}
452 \DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
453 \DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
454 \DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
455 \DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
456 \DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
457 \DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}
458 \DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
459 \DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
460 \DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
461 \DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
462 \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.

```

463 \DeclareRobustCommand
464 \cong{\mathrel{\mathpalette\@ vereq\sim}} % congruence sign
465 \def\@ vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
466 \ialign{$\m@th#1\hfil##\hfil$\crcr#2\crcr=\crcr}}}
467 \DeclareRobustCommand
468 \notin{\mathrel{\m@th\mathpalette\c@ncel\in}}
469 \def\c@ncel#1#2{\m@th\oalign{$\hfil#1\mkern1mu/\hfil$\crcr$#1#2$}}
470 \DeclareRobustCommand
471 \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
472 \def\rlh@#1{\vcenter{\m@th\hbox{\oalign{\raise2pt
473 \hbox{$\#1\rightthreetimesup$}\crcr
474 $\#1\leftharpoondown$}}}}
475 \DeclareRobustCommand
476 \doteq{\buildrel{\textstyle.\over=}}

```

### 6.4.6 Arrows

```
477 \DeclareRobustCommand
478 \joinrel{\mathrel{\mkern-3mu}}
```

```

479 \DeclareRobustCommand
480 \relbar{\mathrel{\smash-}} % \smash, because -
481 % 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 `\Rrightarrow` 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).

```

482 \DeclareRobustCommand
483 \Relbar{\mathrel{=}}
484 \DeclareMathSymbol{\lhook}{\mathrel}{letters}{`2C}
485 \ DeclareRobustCommand\hookrightarrow{\lhook\joinrel\rarrow}
486 \DeclareMathSymbol{\rhook}{\mathrel}{letters}{`2D}
487 \ DeclareRobustCommand\hookleftarrow{\leftarrow\joinrel\rhook}
488 \DeclareRobustCommand
489 \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
490 \DeclareRobustCommand
491 \models{\mathrel{|}\joinrel\Relbar}
492 \DeclareRobustCommand
493 \Longrightarrow{\Relbar\joinrel\Rrightarrow}

```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make them robust.

```

494 \DeclareRobustCommand\longrightarrow
495 {\relbar\joinrel\rarrow}
496 \DeclareRobustCommand\longleftarrow
497 {\leftarrow\joinrel\relbar}
498 \DeclareRobustCommand
499 \Longleftarrow{\Leftarrow\joinrel\Relbar}
500 \DeclareRobustCommand
501 \longmapsto{\mapstochar\longrightarrow}
502 \DeclareRobustCommand
503 \longleftrightarrow{\leftarrow\joinrel\rarrow}
504 \DeclareRobustCommand
505 \Longleftrightarrow{\Leftarrow\joinrel\Rrightarrow}
506 \DeclareRobustCommand
507 \iff{\;}{\Longleftrightarrow\;}

```

#### 6.4.7 Punctuation symbols

```

508 \DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{`3A}
509 \DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{`01}
510 \DeclareMathSymbol{\colon}{\mathpunct}{operators}{`3A}

```

This is commented out, since `\ldots` is now defined in `ltoutenc.dtx`.

```

511 %\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
512 %\ DeclareRobustCommand\ldots
513 % {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots$}\fi}
514 \ DeclareRobustCommand
515 \cdots{\mathinner{\cdotp\cdotp\cdotp}}
516 \ DeclareRobustCommand
517 \vdots{\vbox{\baselineskip4\p@\lineskiplimit\z@}}

```

```

518 \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
519 \DeclareRobustCommand
520 \ddots{\mathinner{\mkern1mu\raise7\p@
521 \vbox{\kern7\p@\hbox{.}}\mkern2mu
522 \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}

```

#### 6.4.8 Math accents

```

523 \DeclareMathAccent{\acute}{\mathalpha}{operators}{13}
524 \DeclareMathAccent{\grave}{\mathalpha}{operators}{12}
525 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{7F}
526 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{7E}
527 \DeclareMathAccent{\bar}{\mathalpha}{operators}{16}
528 \DeclareMathAccent{\breve}{\mathalpha}{operators}{15}
529 \DeclareMathAccent{\check}{\mathalpha}{operators}{14}
530 \DeclareMathAccent{\hat}{\mathalpha}{operators}{5E}
531 \DeclareMathAccent{\vec}{\mathord}{letters}{7E}
532 \DeclareMathAccent{\dot}{\mathalpha}{operators}{5F}
533 \DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{65}
534 \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:

```
535 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{17}
```

#### 6.4.9 Radicals

```
536 \DeclareMathRadical{\sqrtsign}{symbols}{70}{largesymbols}{70}
```

#### 6.4.10 Over and under something, etc

```

537 \DeclareRobustCommand\overrightarrow[1]{\vbox{\m@th\ialign{##\crcr
538 \rightarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}
539 $ \hfil\displaystyle{#1} \hfil$ \crcr}}
540 \DeclareRobustCommand\overleftarrow[1]{\vbox{\m@th\ialign{##\crcr
541 \leftarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}%
542 $ \hfil\displaystyle{#1} \hfil$ \crcr}}
543 \DeclareRobustCommand\overbrace[1]
544 {\mathop{\vbox{\m@th\ialign{##\crcr\noalign{\kern3\p@}%
545 \downbracefill\crcr\noalign{\kern3\p@\nointerlineskip}%
546 $ \hfil\displaystyle{#1} \hfil$ \crcr}}}\limits}
547 \DeclareRobustCommand\underbrace[1]{\mathop{\vtop{\m@th\ialign{##\crcr
548 $ \hfil\displaystyle{#1} \hfil$ \crcr
549 \noalign{\kern3\p@\nointerlineskip}%
550 \upbracefill\crcr\noalign{\kern3\p@}}}\limits}

```

(quite a waste of tokens, IMHO — Frank)

```

551 \DeclareRobustCommand\skew[3]
552 {{\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
553 #2\{\mkern-\muskip\z@#3\mkern\muskip\z@\mkern-\muskip\z@{}\}}
554 \DeclareRobustCommand\rightarrowfill{$\m@th\smash{-\mkern-7mu}%
555 \cleaders\hbox{$\mkern-2mu\smash{-\mkern-2mu}$}\hfill
556 \mkern-7mu\mathord\rightarrow$}
557 \DeclareRobustCommand\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
558 \cleaders\hbox{$\mkern-2mu\smash{-\mkern-2mu}$}\hfill
559 \mkern-7mu\smash-$}
560 \DeclareMathSymbol{\braceleft}{\mathord}{largesymbols}{7A}
561 \DeclareMathSymbol{\braceright}{\mathord}{largesymbols}{7B}

```

```

562 \DeclareMathSymbol{\braceleft}{\mathord}{largesymbols}{7C}
563 \DeclareMathSymbol{\braceright}{\mathord}{largesymbols}{7D}
564 \ DeclareRobustCommand\downbracefill{$\m@th \setbox\z@\hbox{$\braceleft$}%
565 \braceleft\leaders\vrule \height\ht\z@ \depth\z@\hfill\braceright
566 \braceleft\leaders\vrule \height\ht\z@ \depth\z@\hfill\braceright$}%
567 \ DeclareRobustCommand\upbracefill{$\m@th \setbox\z@\hbox{$\braceright$}%
568 \braceright\leaders\vrule \height\ht\z@ \depth\z@\hfill\braceright
569 \braceright\leaders\vrule \height\ht\z@ \depth\z@\hfill\braceright$}

```

#### 6.4.11 Delimiters

```

570 \DeclareMathDelimiter{\lmooustache} % top from (, bottom from)
571 {\mathopen}{largesymbols}{7A}{largesymbols}{40}
572 \DeclareMathDelimiter{\rmoustache} % top from), bottom from (
573 {\mathclose}{largesymbols}{7B}{largesymbols}{41}
574 \DeclareMathDelimiter{\arrowvert} % arrow without arrowheads
575 {\mathord}{symbols}{6A}{largesymbols}{3C}
576 \DeclareMathDelimiter{\Arrowvert} % double arrow without arrowheads
577 {\mathord}{symbols}{6B}{largesymbols}{3D}
578 \DeclareMathDelimiter{\Vert}
579 {\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

```

580 \%let\|=\\Vert
581 \DeclareMathDelimiter{\|}{%
582 {\mathord}{symbols}{6B}{largesymbols}{0D}}
583 \DeclareMathDelimiter{\vert}{%
584 {\mathord}{symbols}{6A}{largesymbols}{0C}}
585 \DeclareMathDelimiter{\uparrow}{%
586 {\mathrel}{symbols}{22}{largesymbols}{78}}
587 \DeclareMathDelimiter{\downarrow}{%
588 {\mathrel}{symbols}{23}{largesymbols}{79}}
589 \DeclareMathDelimiter{\updownarrow}{%
590 {\mathrel}{symbols}{6C}{largesymbols}{3F}}
591 \DeclareMathDelimiter{\Uparrow}{%
592 {\mathrel}{symbols}{2A}{largesymbols}{7E}}
593 \DeclareMathDelimiter{\Downarrow}{%
594 {\mathrel}{symbols}{2B}{largesymbols}{7F}}
595 \DeclareMathDelimiter{\Updownarrow}{%
596 {\mathrel}{symbols}{6D}{largesymbols}{77}}
597 \DeclareMathDelimiter{\backslash}{% for double coset G\backslash H
598 {\mathord}{symbols}{6E}{largesymbols}{0F}}
599 \DeclareMathDelimiter{\rangle}{%
600 {\mathclose}{symbols}{69}{largesymbols}{0B}}
601 \DeclareMathDelimiter{\langle}{%
602 {\mathopen}{symbols}{68}{largesymbols}{0A}}
603 \DeclareMathDelimiter{\rbrace}{%
604 {\mathclose}{symbols}{67}{largesymbols}{09}}
605 \DeclareMathDelimiter{\lbrace}{%
606 {\mathopen}{symbols}{66}{largesymbols}{08}}
607 \DeclareMathDelimiter{\rceil}{%
608 {\mathclose}{symbols}{65}{largesymbols}{07}}
609 \DeclareMathDelimiter{\lceil}{%
610 {\mathopen}{symbols}{64}{largesymbols}{06}}

```

```

611 \DeclareMathDelimiter{\rfloor}
612 {\mathclose}{symbols}{63}{largesymbols}{05}
613 \DeclareMathDelimiter{\lfloor}
614 {\mathopen}{symbols}{62}{largesymbols}{04}

\lgroup There are three plain TeX delimiters which are not fully supported by NFSS, since they
\rgroup partly point into a bold cmr font. Allocating a full symbol font, just to have three
\bracevert 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.)
615 \DeclareMathDelimiter{\lgroup} % extensible (with sharper tips
616 {\mathopen}{largesymbols}{3A}{largesymbols}{3A}
617 \DeclareMathDelimiter{\rgroup} % extensible) with sharper tips
618 {\mathclose}{largesymbols}{3B}{largesymbols}{3B}
619 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
620 {\mathord}{largesymbols}{3E}{largesymbols}{3E}

```

(End of definition for `\lgroup`, `\rgroup`, and `\bracevert`.)

## 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 These math symbols are not in plain TeX.
\mathsection
\mathdollar
\mathsterling
\mathunderscore
621 \DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{7B}
622 \DeclareMathSymbol{\mathsection}{\mathord}{symbols}{78}
623 \DeclareMathSymbol{\mathdollar}{\mathord}{operators}{24}
624 \DeclareRobustCommand{\mathsterling}{\mathit{\mathchar"7024}}
625 \DeclareRobustCommand{\mathunderscore}{\kern.06em\vbox{\hrule\@width.3em}}

```

(End of definition for `\mathparagraph` and others.)

`\mathellipsis` This is plain TeX's `\ldots`.

```
626 \DeclareRobustCommand{\mathellipsis}{\mathinner{\ldotp\ldotp\ldotp}}%
```

(End of definition for `\mathellipsis`.)

## 6.6 Other special functions and parameters

### 6.6.1 Biggggg

```

627 </math>
628 <*math | latexrelease>
629 <latexrelease>\IncludeInRelease{2018/12/01}%
630 <latexrelease> {\Big}{Start LR-mode}%
631 \DeclareRobustCommand{\big[1]{\leavevmode@ifvmode
632 {\hbox{$\left.\vphantom{\big|}\right.^8.5\p@{}\right.\n@space$}}}}
633 \DeclareRobustCommand{\Big[1]{\leavevmode@ifvmode
634 {\hbox{$\left.\vphantom{\big|}\right.^{11.5\p@{}\right.\n@space$}}}}

```

```

635 \DeclareRobustCommand\bigg[1]{\leavevmode@ifvmode
636 {\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
637 \DeclareRobustCommand\Bigg[1]{\leavevmode@ifvmode
638 {\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
639 {/math | \textralate{release}}
640 {|\textralate{release}|\EndIncludeInRelease
641 {|\textralate{release}|}\IncludeInRelease{0000/00/00}%
642 {|\textralate{release}|} {\Big}{Start LR-mode}%
643 {|\textralate{release}|}\def\big#1{\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
644 {|\textralate{release}|}\def\Big#1{\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
645 {|\textralate{release}|}\def\bigg#1{\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
646 {|\textralate{release}|}\def\Bigg#1{\hbox{$\left.\vphantom{\bigg|}\right.\vphantom{\bigg|}$}}\right.\n@space$}}}
647 {|\textralate{release}|}\EndIncludeInRelease
648 {*math}
649 \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.

```
650 \def\operator@font{\mathgroup\symoperators}
```

(End of definition for \operator@font.)

### 6.6.3 Parameters

```

651 \thinmuskip=3mu
652 \medmuskip=4mu plus 2mu minus 4mu
653 \thickmuskip=5mu plus 5mu

```

This finishes the low-level setup in `fontmath.ltx`.

```
654 {/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.

```

655 {*cfgtext | cfgmath | cfgprel}
656 %%
657 %%
658 %%
659 %% Load the standard setup:
660 %%
661 {+cfgtext}\input{fonttext.ltx}
662 {+cfgmath}\input{fontmath.ltx}
663 {+cfgprel}\input{preload.ltx}
664 %%
665 %% Small changes could go here; see documentation in cfgguide.tex for
666 %% allowed modifications.
667 %%
668 %% In particular it is not allowed to misuse this configuration file
669 %% to modify internal LaTeX commands!
670 %%
671 %% If you use this file as the basis for configuration please change
672 %% the \ProvidesFile lines to clearly identify your modification, e.g.,
673 %%

```

```
674 <+cfgtext>%% \ProvidesFile{fonttext.cfg}[2001/06/01
675 <+cfgmath>%% \ProvidesFile{fonttext.cfg}[2001/06/01
676 <+cfgprel>%% \ProvidesFile{preload.cfg}[2001/06/01
677 %% Customised local font setup]
678 %%
679 %%
680 </cfgtext | cfgmath | cfgprel>
```

# File 31

## preload.dtx

### 1 Overview

This file contains a number of possible settings for preloading fonts during installation of NFSS2 (which is used by L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> ). It will be used to generate the following files:

|               |                                                                |
|---------------|----------------------------------------------------------------|
| preload.min   | minimal subset of fonts necessary to run NFSS2                 |
| preload.ori   | preload of CM fonts similar to the old <code>1fonts.tex</code> |
| preload.ltx   | The standard selection of preloads                             |
| cmpreload.xpt | preload of CM fonts for 10pt document size                     |
| cmpreload.xip | preload of CM fonts for 11pt document size                     |
| cmpreload.xii | preload of CM fonts for 12pt document size                     |
| dcpreload.xpt | preload of DC fonts for 10pt size                              |
| dcpreload.xip | preload of DC fonts for 11pt size                              |
| dcpreload.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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  system by installing a file with the name `preload.cfg`. If this file exists it will be used in place of the system file `preload.ltx`. You can, for example, copy one of the files mentioned above (that can be generated from this source) to `preload.cfg`.

Or you can define completely other preloads. In that case start from `preload.min` since that contains the fonts that have to be preloaded by \*all\* L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  systems.

Avoid using `preload.ori`, it will load so many fonts that on most installations it is nearly impossible to load other font families afterwards. This file is only generated to show what fonts have been preloaded by L<sup>A</sup>T<sub>E</sub>X 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 <code>lfonts.tex</code> |
| 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  $\text{\TeX}$ , i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 {*driver}
2 \documentclass{ltxdoc}
3 %\OnlyDescription % comment out for implementation details
4 \begin{document}
5 \DocInput{preload.dtx}
6 \end{document}
7
```

## 5 The code

We begin by loading the math extension font (`cmex10`) and the  $\text{\LaTeX}$  line and circle fonts. It is necessary to do this explicitly since these are used by the  $\text{\LaTeX}$  format. Since the internal font name contains / characters and digits we construct the name via `\csname`. These are the only fonts (!) that must be loaded in this file.

All `\DeclarePreloadSizes` can be removed or others can be added, they only influence the processing speed.

```
8 \expandafter\font\csname OMX/cmex/m/n/10\endcsname=cmex10\relax
9 \font\tenln =line10 \font\tenlnw =linew10\relax
10 \font\tencirc=lcircle10 \font\tencircw=lcirclew10\relax
```

The above fonts should not be touched but anything below this point here in the preload suggestions can be modified without any problems.

```
11 {-tex}*****%
12 {-tex}% Start any modification below this point **
13 {-tex}*****%
14 {-tex}%
15 %%%
16 %% Computer Modern Roman:
17 %%-----
```

```

18 <*ori>
19 \DeclarePreloadSizes{OT1}{cmr}{m}{n}
20 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
21 \DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
22 \DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
23 \DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
24 </ori>
25 <+xpt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
26 <+xpt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
27 <+xipt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
28 <+xipt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
29 <+xiipt & cm> \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
30 <+xiipt & dc> \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
31 %%
32 %% Computer Modern Sans:
33 %-----%
34 <+ori> \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
35 %%
36 %% Computer Modern Typewriter:
37 %-----%
38 <+ori> \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
39 %%
40 %% Computer Modern Math:
41 %-----%
42 <*ori>
43 \DeclarePreloadSizes{OML}{cmm}{m}{it}
44 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
45 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}
46 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
47 </ori>

```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```

48 <*xpt>
49 \DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
50 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
51 </xpt>
52 <*xipt>
53 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
54 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
55 </xipt>
56 <*xiipt>
57 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
58 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
59 </xiipt>
60 %%
61 %% LaTeX symbol fonts:
62 %-----%
63 <*ori>
64 \DeclarePreloadSizes{U}{lasy}{m}{n}
65 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
66 </ori>
67 </preload>

```

## File 32

# ltfntcmd.dtx

### Abstract

The commands defined in this file `ltfntcmd` are part of the kernel code for L<sup>A</sup>T<sub>E</sub>X 2<sub>&</sub>/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 T<sub>E</sub>X system and for several reasons it is better to avoid them on the user level whenever possible. In L<sup>A</sup>T<sub>E</sub>X3 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 3 shows all these high-level commands in action. A further advantage of using these commands is that they automatically take care of any necessary italic correction on either side of their argument.

Thus, when using such commands, one does not have to worry about forgetting the italic correction when changing fonts. Only in very few situations is this additional space wrong but, for example, most typographers recommend omitting the italic correction if a small punctuation character, like a comma, directly follows the font change. Since the amount of correction required is partly a matter of taste, you can define in what situations the italic correction should be suppressed. This is done by putting the characters that should cancel a preceding italic correction in the list `\nocorrlist`.<sup>38</sup> 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
```

---

<sup>38</sup>Any package that changes the `\catcode` of a character inside `\nocorrlist` must then explicitly reset the list. Otherwise the changed character will no longer be recognized by the suppression algorithm.

| <i>Command</i>               | <i>Corresponds to</i>    | <i>Action</i>                                      |
|------------------------------|--------------------------|----------------------------------------------------|
| <code>\textnormal{..}</code> | <code>\normalfont</code> | Typeset argument in normal family                  |
| <code>\textrm{..}</code>     | <code>\rmfamily</code>   | Typeset argument in roman family                   |
| <code>\textsf{..}</code>     | <code>\sffamily</code>   | Typeset argument in <code>sans serif</code> family |
| <code>\texttt{..}</code>     | <code>\ttfamily</code>   | Typeset argument in <code>typewriter</code> family |
| <code>\textmd{..}</code>     | <code>\mdseries</code>   | Typeset argument in medium series                  |
| <code>\textbf{..}</code>     | <code>\bfseries</code>   | Typeset argument in <b>bold</b> series             |
| <code>\textup{..}</code>     | <code>\upshape</code>    | Typeset argument in normal shape                   |
| <code>\textit{..}</code>     | <code>\itshape</code>    | Typeset argument in <i>italic</i> shape            |
| <code>\textsl{..}</code>     | <code>\slshape</code>    | Typeset argument in <i>slanted</i> shape           |
| <code>\textsc{..}</code>     | <code>\scshape</code>    | Typeset argument in <code>SMALL CAPS</code> shape  |
| <code>\emph{..}</code>       | <code>\em</code>         | Typeset argument <i>emphasized</i>                 |

Table 3: 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 L<sup>A</sup>T<sub>E</sub>X by adding a `\nocorr` command.*

In contrast, the use of the declaration forms is often more appropriate when you define your own commands or environments.

```
\newenvironment{bfitemize}{\begin{itemize}\normalfont\bfseries}
{\end{itemize}}
\begin{bfitemize}
\item This environment produces boldface items.
\item It is defined in terms of \LaTeX's
 \texttt{itemize} environment and NFSS
 declarations.
\end{bfitemize}
```

This gives:

- This environment produces boldface items.
- It is defined in terms of L<sup>A</sup>T<sub>E</sub>X's `itemize` environment and NFSS declarations.

In addition to global customization of when to insert the italic correction, it is of course sometimes necessary to explicitly insert one with `\!/`.

It is also possible to suppress the italic correction in individual instances. For this, the command `\nocorr` is provided.

The `\nocorr` must appear as the first or last token inside the braces of the argument of the `\text...` commands, at that end of the text where you wish to suppress the italic correction.

It is worth pointing out here that inserting a `\V` 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 `\V` in `amstex.sty` is used as this version can cause space to be removed immediately before the `\V`.

## 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 optimizes this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other aftergroup tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```
1 {*2ekernel}
2 \def \DeclareTextFontCommand #1#2{%
3 \DeclareRobustCommand#1[1]{%
4 \ifmmode
5 \nfss@text{#2##1}%
6 \else
7 \hmode@bgroup
8 \text@command{##1}%
9 #2\check@icl ##1\check@icr
10 \expandafter
11 \egroup
12 \fi
13 }%
14 }
```

(End of definition for `\DeclareTextFontCommand`.)

`\textrm` Now we define the `\text<family>` commands in terms of the above; `\textttt` does not look very nice!

`\textsf` 15 `\DeclareTextFontCommand{\textrm}{\rmfamily}`  
`\textnormal` 16 `\DeclareTextFontCommand{\textsf}{\sffamily}`  
17 `\DeclareTextFontCommand{\textttt}{\ttfamily}`  
18 `\DeclareTextFontCommand{\textnormal}{\normalfont}`

(End of definition for `\textrm` and others.)

`\textbf` For the series attribute:

`\textmd` 19 `\DeclareTextFontCommand{\textbf}{\bfseries}`  
20 `\DeclareTextFontCommand{\textmd}{\mdseries}`

(End of definition for `\textbf` and `\textmd`.)

`\textit` And for the shapes:

`\textsl` 21 `\DeclareTextFontCommand{\textit}{\itshape}`  
`\textsc` 22 `\DeclareTextFontCommand{\textsl}{\slshape}`  
`\textup` 23 `\DeclareTextFontCommand{\textsc}{\scshape}`  
24 `\DeclareTextFontCommand{\textup}{\upshape}`

(End of definition for `\textit` and others.)

`\textulc`  
`\textsw` 25 `/2ekernel`  
`\textssc` 26 `{*2ekernel | latexrelease}`  
27 `\langle latexrelease \rangle \IncludeInRelease{2020/02/02} %`  
28 `\langle latexrelease \rangle \textulc \AdditionalTextCommands %`  
29 `\DeclareTextFontCommand{\textulc}{\ulcshape}`  
30 `\DeclareTextFontCommand{\textsw}{\swshape}`  
31 `\DeclareTextFontCommand{\textssc}{\sscshape}`  
32 `\langle latexrelease \rangle /2ekernel | latexrelease`  
33 `\langle latexrelease \rangle \EndIncludeInRelease`  
34 `\langle latexrelease \rangle \IncludeInRelease{0000/00/00} %`  
35 `\langle latexrelease \rangle \textulc \AdditionalTextCommands %`  
36 `\langle latexrelease \rangle`  
37 `\langle latexrelease \rangle \let \textulc @undefined`  
38 `\langle latexrelease \rangle \let \textsw @undefined`  
39 `\langle latexrelease \rangle \let \textssc @undefined`  
40 `\langle latexrelease \rangle \EndIncludeInRelease`  
41 `{*2ekernel}`

(End of definition for `\textulc`, `\textsw`, and `\textssc`.)

`\emph` Finally we have the `\em` font change declaration of L<sup>A</sup>T<sub>E</sub>X. The corresponding definition with argument is

42 `\DeclareTextFontCommand{\emph}{\em}`

(End of definition for `\emph`.)

`\nocorr` This is just a label, so it does nothing; it should also be unexpandable.

43 `\let \nocorr \relax`

(End of definition for \nocorr.)

- \check@ic1 We define these defaults in case some error causes them to be expanded at the wrong time.  
\check@icr

```
44 \let \check@ic1 \@empty
45 \let \check@icr \@empty
```

(End of definition for \check@ic1 and \check@icr.)

- \text@command This checks for a \nocorr as the first token in its argument and also for one in any other position not protected within braces (the latter is treated as if it were at the end of the argument).

Is this the correct action in the ‘empty’ case? It is efficient but typographically it is, strictly, incorrect!

```
46 \def \text@command #1{
47 \edef \reserved@a {\unexpanded{#1}}%
48 \ifx \reserved@a \@empty
49 \let \check@ic1 \@empty
50 \let \check@icr \@empty
51 \else
```

\space is a reserved word in L<sup>A</sup>T<sub>E</sub>X or actually already in plain T<sub>E</sub>X. If somebody really redefines it so many things will break that I don’t see any reason to make this routine here slower than necessary.

```
52 % \def \reserved@b { }%
53 % \ifx \reserved@a \reserved@b
54 \ifx \reserved@a \space
55 \let \check@ic1 \@empty
56 \let \check@icr \@empty
57 \else
58 \check@nocorr@ #1\nocorr@nil
59 \fi
60 \fi
61 }
62 \def \check@nocorr@ #1#2\nocorr#3@nil {%
```

The two checks are initialised here to their values in the normal case.

```
63 \let \check@ic1 \maybe@ic
64 \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi} %
65 \def \reserved@a {\nocorr}%
66 \def \reserved@b {#1}%
67 \def \reserved@c {#3}%
68 \ifx \reserved@a \reserved@b
69 \ifx \reserved@c \@empty
```

In this case there is a \nocorr at the start but not at the end, so \check@ic1 should be empty.

```
70 \let \check@ic1 \@empty
71 \else
```

Otherwise there is a \nocorr both at the start and elsewhere, so no italic corrections should be added.

```
72 \let \check@ic1 \@empty
73 \let \check@icr \@empty
74 \fi
```

```

75 \else
76 \ifx \reserved@c \empty
```

In this case there is no `\nocorr` anywhere, so we need to check for an italic correction at both the beginning and the end. This has been set up as the default so no code is needed here.

```

77 \else
```

In this case there is no `\nocorr` at the start but there is one elsewhere, so no `\aftergroup` is needed.

```

78 \let \check@icr \empty
79 \fi
80 \fi
81 }
```

*(End of definition for `\text@command` and `\check@nocorr@`.)*

`\ifmaybe@ic` Switch used solely within `\maybe@ic` not interfering with other switches.

```

82 \newif\ifmaybe@ic
```

*(End of definition for `\ifmaybe@ic`.)*

`\maybe@ic` These macros implement the italic correction.

```

\maybe@ic@ 83 \def \maybe@ic {\futurelet\@let@token\maybe@ic@}
84 \def \maybe@ic@ {\%
```

We first check to see if the current font is positively sloped. (But do not forget the message Rainer sent about an upright font with non-zero slope! Or is this an urban myth?) It has been suggested that this should test against a small positive value, but what?

```

85 \ifdim \fontdimen@ne\font>\z@
86 \else
87 \maybe@ictrue
```

It would be possible, but probably not worthwhile, to continue the forward scan beyond any closing braces.

```

88 \expandafter\@tfor\expandafter\reserved@a\expandafter:\expandafter=%
89 \nocorrlist
```

We have to hide the `\@let@token` in the macro `\t@st@ic` rather than testing it directly in the loop since it might be `\let` to a `\fi` or `\else`, which would result in chaos.

```

90 \do \t@st@ic
```

Frank thinks that the next bit is inefficient if done after the second change. Chris thinks that most all of this is inefficient for the commonest cases: but that is the price of a cleverer algorithm. It is certainly needed to deal with the use of `\nolinebreak`.

```

91 \ifmaybe@ic \sw@slant \fi
92 \fi
93 }
```

*(End of definition for `\maybe@ic` and `\maybe@ic@`.)*

`\t@st@ic` The next token in the input stream is stored in `\@let@token` via a `\let`, the current token from `\nocorrlist` is stored via `\def` in `\reserved@a`. To compare them we have to fiddle around a bit.

If the only things to check were characters then this could be done via an `\if` thus their catcodes would not matter; but this will not work whilst `\futurelet` is used above.

```

94 \def \t@st@ic {%
95 \expandafter\let\expandafter\reserved@b\expandafter=\reserved@a\relax
96 \ifx\reserved@b\@let@token

```

If they are the same we record the fact and jump out of the loop.

```

97 \maybe@icfalse
98 \@break@tfor
99 \fi
100 }

```

(End of definition for `\t@st@ic`.)

`\sw@slant` The definition of the mysterious `\sw@slant` command is as follows.

```

101 \def \sw@slant {%

```

It is surely correct to put in an italic correction when there is no skip. If the last thing on the list is actually a zero skip (including things whose dimension part is zero, such as `\hfill`), or anything other than a character, then the italic correction will have no effect.

In order to work correctly with unbreakable spaces from `\~` (and other common forms of line-breaking control) we also move back across a penalty before the glue.

```

102 \ifdim \lastskip=\z@
103 \fix@penalty
104 \else
105 \skip@\lastskip
106 \unskip
107 \fix@penalty
108 \hskip \skip@
109 \fi
110 }

```

The above code means: “If there is a non-zero space just before the current position (`\ifdim...`) save the amount of that space (`\skip@\lastskip`), remove it (`\unskip`), then do a similar thing if there is a penalty just before the skip, and finally put the space back in.”

Since zero glue cannot be distinguished in this context from no glue, we dare not put in an `\hskip` in this case as this may produce an unwanted breakpoint. This is not satisfactory.

The penalty before the glue is handled similarly, with the same caveats concerning the zero case. Is this the first recorded use of `\unpenalty` in standard L<sup>A</sup>T<sub>E</sub>X code?

```

111 \def \fix@penalty {%
112 \ifnum \lastpenalty=\z@
113 \@@italiccorr
114 \else
115 \count@\lastpenalty
116 \unpenalty
117 \@@italiccorr

```

```

118 \penalty \count0
119 \fi
120 }

```

(End of definition for `\sw@slant` and `\fix@penalty`.)

- `\nocorrlist` This holds the list of characters that should prevent italic correction. They should be ordered by decreasing frequency of use. If any such character is made active later on one needs to redefine the list so that the active character becomes part of it.

```

121 \def \nocorrlist {,.}

```

(End of definition for `\nocorrlist`.)

- `\nfss@text` This command will by default behave like a L<sup>A</sup>T<sub>E</sub>X `\mbox` but may be redefined by packages such as `amstext.sty` to be a bit cleverer.

```

122 \ifx \nfss@text \undefined
123 \def \nfss@text {\leavevmode\hbox}
124 \fi

```

(End of definition for `\nfss@text`.)

- `\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\sfamily}{\mathrm{sf}}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathrm{tt}}

```

```

125 \def \DeclareOldFontCommand #1#2#3{%
126 \ DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
127 }

```

(End of definition for `\DeclareOldFontCommand`.)

- `\@fontswitch` These two commands actually do the necessary tests and declarative font- or alphabet-changing.

```

128 \def \@fontswitch #1#2{%
129 \ifmmode
130 \let \math@bgroup \relax
131 \def \math@egroup {\let \math@bgroup \@@math@bgroup
132 \let \math@egroup \@@math@egroup}%

```

We need to have a `\relax` in the following line in case the #2 is something like `\mathsf` grabbing the next token as an argument. For this reason the code also uses explicit arguments again (see pr/1275).

```
133 #2\relax
134 \else
135 #1%
136 \fi
137 }
138 \let \@@math@bgroup \math@bgroup
139 \let \@@math@egroup \math@egroup

(End of definition for \fontswitch, \math@bgroup, and \math@egroup.)
These commands are available only in the preamble.

140 \onlypreamble \DeclareTextFontCommand
141 \onlypreamble \DeclareOldFontCommand
```

### 3 Initialization

`\normalsize` This is defined to produce an error.

```
142 \def\normalsize{%
143 \@latex@error {The font size command \protect\normalsize\space
144 is not defined:\MessageBreak
145 there is probably something wrong with
146 the class file}\@eha
147 }
148 ⟨/2ekernel⟩

(End of definition for \normalsize.)
```

## File 33

# 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).

```
1 {*2ekernel | latexrelease}
2 <latexrelease>\NewModuleRelease{2020/02/02}{lttextcomp}
3 <latexrelease> {Text Companion symbols}
```

\oldstylenums 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.

```
4 \DeclareRobustCommand\legacyoldstylenums[1]{%
5 \begingroup
```

Provide spacing using the interword space of the current font.

```
6 \spaceskip\fontdimen\tw@\font
```

Then switch to the math italic font. We don't change the current value of `\f@series` which means that you can use bold numerals if `\bfseries` is in force. As family we use `\rmdefault` which means that this only works if there exist an OML encoded version of that font or rather a corresponding .fd file (which is the case for standard L<sup>A</sup>T<sub>E</sub>X fonts even though they only contain substitutions).

```
7 \usefont{OML}{\rmdefault}{\f@series}{it}%
8 \mathgroup\symletters #1%
9 \endgroup
10 }
```

And here is the improved one that adjusts depending on surroundings.

```
11 \DeclareRobustCommand\oldstylenums[1]{%
12 \begingroup
13 \ifmmode
14 \mathgroup\symletters #1%
15 \else
```

The `\CheckEncodingSubset` is discussed below.

```
16 \CheckEncodingSubset\use@text@encoding{TS1}\tc@oldstylesubst2{{#1}}%
17 \fi
18 \endgroup
19 }
```

The helper to select the substitution if needed.

```
20 \def\tc@oldstylesubst#1{%
21 \tc@errorwarn
22 {Oldstyle digits unavailable for
23 family \f@family.\MessageBreak
24 Default oldstyle digits used instead}\@eha
25 \bgroup
26 \expand@font@defaults
```

The substitution defaults are provided in the file `fonttext.ltx`.

```
27 \ifx\f@family\rmdef@ult
28 \fontfamily\rmsubstdefault
29 \else\ifx\f@family\sff@ult
30 \fontfamily\sfsbstdefault
31 \else\ifx\f@family\ttdef@ult
32 \fontfamily\ttsbstdefault
33 \else
34 \fontfamily\textcompsubstdefault
35 \fi\fi\fi
36 \fontencoding{TS1}\selectfont#1%
37 \egroup
38 }
```

(End of definition for `\oldstylenums` and `\legacyoldstylenums`.)

`\textcompsubstdefault` Here is the default for the “unknown” case:

```
39 \def\textcompsubstdefault{\rmsubstdefault}
```

(End of definition for `\textcompsubstdefault`.)

`\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, with a value between 0 (all of the encoding is supported) and 9 (many glyphs are missing).

For TS1 the numbers have been chosen in a way that most fonts can be fairly correctly categorized, but the default settings are always conservative, that is they may claim that less glyphs are supported than there actually are.

As these days many font families are set up to end in -LF (lining figures), -OsF (oldstyle figures), etc. the declaration supports a shortcut: if the *font family* name ends in -\* then the star gets replaced by these common ending, e.g.,

```
\DeclareEncodingSubset{TS1}{Alegreya-*}{2}
```

is the same as writing

```
\DeclareEncodingSubset{TS1}{Alegreya-LF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-OsF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-TLF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-T0sF}{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.

```
40 \def\DeclareEncodingSubset#1#2{%
41 \DeclareEncodingSubset@aux{#1}#2*\ DeclareEncodingSubset@aux
42 }
43 \def\DeclareEncodingSubset@aux#1#2*#3\DeclareEncodingSubset@aux#4{%
```

if #3 is empty then there was no star, otherwise we define all four variants.

```
44 \expandafter\ifx\expandafter X\detokenize{#3}X%
45 \@DeclareEncodingSubset{#1}{#2}{#4}%
46 \else
47 \@DeclareEncodingSubset{#1}{#2LF}{#4}%
48 \@DeclareEncodingSubset{#1}{#2TLF}{#4}%
49 \@DeclareEncodingSubset{#1}{#20sF}{#4}%
50 \@DeclareEncodingSubset{#1}{#2T0sF}{#4}%
51 \fi
52 }
```

The subset info is stored in a command with the name `\family:subset` so if that already exists we change otherwise declare a subset.

```
53 \def\@DeclareEncodingSubset#1#2#3{%
54 \@ifundefined{#1:#2}{%
55 {\@font@info{Setting #2 sub-encoding to #1/#3}}%
56 {\@font@info{Changing #2 sub-encoding to #1/#3}}%
```

This declaration should be usable in `.fd` files and therefore has to make its definition globally, because such files can get loaded in random places.

```
57 \global\@namedef{#1:#2}{#3}%
58 Any reason to allow those in the middle of documents?
59 \onlypreamble\DeclareEncodingSubset
60 \onlypreamble\DeclareEncodingSubset@aux
61 \onlypreamble\@DeclareEncodingSubset
```

(End of definition for `\DeclareEncodingSubset`.)

`\CheckEncodingSubset` The command `\CheckEncodingSubset` will check if the current font family has the right encoding subset to typeset a certain command. It takes five arguments as follows: first argument is either `\UseTextSymbol`, `\UseTextAccent` depending on whether or not the symbol is a text symbol or a text accent.

The second argument is the encoding from which this symbol should be fetched.

The third argument is either a fake accessor command or an error message. the code in that argument (if ever executed) receives two arguments: #2 and #5 of `\CheckEncodingSubset`.

Argument four is the subset encoding id to test against: if this value is higher than the subset id of the current font family then we typeset the symbol, i.e., execute #1{#2}#5 otherwise it runs #3#5, e.g., to produce an error message or fake the glyph somehow.

Argument five is the symbol or accent command that is being checked.

For usage examples see definitions below.

```
61 \def\CheckEncodingSubset#1#2#3#4#5{%
62 \ifnum #4>%
63 \expandafter\ifx\csname #2:\f@family\endcsname\relax
64 \csname #2:?\endcsname
65 \else
66 \csname #2:\f@family\endcsname
67 \fi
68 \relax
69 \expandafter\@firstoftwo
70 \else
71 \expandafter\@secondoftwo
72 \fi
```

```

73 {#1{#2}}{#3}%
74 #5%
75 }

```

(End of definition for \CheckEncodingSubset.)

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.

```

76 \def\tc@errorwarn#1#2{\@latex@info{#1}}

```

(End of definition for \tc@errorwarn.)

\tc@subst

```

77 \def\tc@subst#1{%
78 \tc@errorwarn
79 {Symbol \string#1 not provided by\MessageBreak
80 font family \f@family\space
81 in TS1 encoding.\MessageBreak Default family used instead}\@eha
82 \bgroup
83 \expand@font@defaults
84 \ifx\f@family\rmdef@ult
85 \fontfamily\rmsubstdefault
86 \else\ifx\f@family\sfdef@ult
87 \fontfamily\sfsubstdefault
88 \else\ifx\f@family\ttdef@ult
89 \fontfamily\ttsubstdefault
90 \else
91 \fontfamily{textcompsubstdefault}
92 \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.

```

93 \namedef{TS1:\f@family}{0}%
94 \selectfont#1%
95 \egroup
96 }

```

(End of definition for \tc@subst.)

\tc@fake@euro \tc@fake@euro is an example of a “fake” definition to use in arg #3 of the command \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 “=“.

```

97 \def\tc@fake@euro#1{%
98 \leavevmode
99 \@font@info{Faking \noexpand#1 for font family
100 \f@family\MessageBreak in TS1 encoding}%
101 \valign{##\cr
102 \vfil\hbox to 0.07em{\dimen@\f@size\p@
103 \math@fontsfalse
104 \fontsize{.7\dimen@}\z@\selectfont=\hss}%
105 \vfil\cr%
106 \hbox{C}\crcr
107 }%
108 }

```

(End of definition for \tc@fake@euro.)

\tc@check@symbol These are two abbreviations that we use below to check symbols and accents in TS1.  
\tc@check@accent Only there to save some space, e.g., we can then write

```
DeclareTextCommandDefault{\textcurrency}{\tc@check@symbol3\textcurrency}
```

to ensure that \textcurrency is only typeset if the current font has a TS1 subset id of less than 3. Otherwise \tc@error is called telling the user that for this font family \textcurrency is not available.

```
109 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
```

Accents have been made an error in the `textcomp` package when not available. Now that we provide the functionality in the kernel we avoid the error by swapping in a T1 accent if the TS1 accent is not available.

```
110 \%def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
111 \def\tc@check@accent#1{\CheckEncodingSubset\UseTextAccent
112 {TS1}{\tc@swap@accent#1}}
113 \def\tc@swap@accent#1#2{\UseTextAccent{T1}#1}
```

(End of definition for \tc@check@symbol and \tc@check@accent.)

## 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.

```
114 \DeclareTextSymbolDefault{\textdollar}{TS1}
115 \UndeclareTextCommand{\textdollar}{OT1} % don't use the OT1 def any longer
116 \DeclareTextSymbolDefault{\textsterling}{TS1}
117 \UndeclareTextCommand{\textsterling}{OT1} % don't use the OT1 def any longer
118 \DeclareTextSymbolDefault{\textperthousand}{TS1}
119 \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:

```
120 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
121 \DeclareTextSymbolDefault{\textbullet}{TS1}
122 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
123 \DeclareTextSymbolDefault{\textdagger}{TS1}
124 \DeclareTextSymbolDefault{\textparagraph}{TS1}
125 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
126 \DeclareTextSymbolDefault{\textsection}{TS1}
```

And here are the other TS1 glyphs that are implemented by every font (or nearly every)—a few are commented out and moved to sub-encoding 9, because they aren't around in some fonts.

```
127 %%\DeclareTextSymbolDefault{\textbardbl}{TS1} % subst in sub-enc 9 above
128 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
```

```

129 %%\DeclareTextSymbolDefault{\textcelsius}{TS1} % subst in sub-enc 9 above
130 \DeclareTextSymbolDefault{\textcent}{TS1}
131 \DeclareTextSymbolDefault{\textcopyright}{TS1}
132 \DeclareTextSymbolDefault{\textdegree}{TS1}
133 \DeclareTextSymbolDefault{\textdiv}{TS1}
134 \DeclareTextSymbolDefault{\textlnot}{TS1}
135 \DeclareTextSymbolDefault{\textonehalf}{TS1}
136 \DeclareTextSymbolDefault{\textonequarter}{TS1}
137 %%\DeclareTextSymbolDefault{\textonesuperior}{TS1} % subst in sub-enc 9 above
138 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
139 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
140 \DeclareTextSymbolDefault{\textpm}{TS1}
141 \DeclareTextSymbolDefault{\textquotesignle}{TS1}
142 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
143 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
144 \DeclareTextSymbolDefault{\textregistered}{TS1}
145 %%\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1} % subst in sub-enc 9 above
146 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
147 %%\DeclareTextSymbolDefault{\textthreesuperior}{TS1} % subst in sub-enc 9 above
148 \DeclareTextSymbolDefault{\textttimes}{TS1}
149 \DeclareTextSymbolDefault{\texttrademark}{TS1}
150 %%\DeclareTextSymbolDefault{\texttwelveudash}{TS1} % subst in sub-enc 9 above
151 %%\DeclareTextSymbolDefault{\texttwosuperior}{TS1} % subst in sub-enc 9 above
152 \DeclareTextSymbolDefault{\textyen}{TS1}
153 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
154 \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 listed as becoming unavailable in sub-encodings  $x + 1$  and higher.

## 1.1 Unavailable in sub-encoding 1 and higher (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.

```

155 \DeclareTextCommandDefault{\textcircled}
156 {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OMS}}1\textcircled}

```

## 1.2 Unavailable in sub-encoding 2 (majority of new OTF fonts via autoinst) and higher

```

157 \DeclareTextCommandDefault{\t}
158 {\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.

```

159 \ifx\Umathcode\@undefined
160 \DeclareTextCommandDefault{\capitalacute}{\@tc@check@accent{'2}\capitalacute}
161 \DeclareTextCommandDefault{\capitalbreve}{\@tc@check@accent{\u}2\capitalbreve}
162 \DeclareTextCommandDefault{\capitalcaron}{\@tc@check@accent{\v}2\capitalcaron}
163 \DeclareTextCommandDefault{\capitalcedilla}{\@tc@check@accent{\c}2\capitalcedilla}
164 \DeclareTextCommandDefault{\capitalcircumflex}{\@tc@check@accent{\^}2\capitalcircumflex}
165 \DeclareTextCommandDefault{\capitaldieresis}{\@tc@check@accent{"2}\capitaldieresis}
166 \DeclareTextCommandDefault{\capitaldotaccent}{\@tc@check@accent{\.2}\capitaldotaccent}
167 \DeclareTextCommandDefault{\capitalgrave}{\@tc@check@accent{\`2}\capitalgrave}
168 \DeclareTextCommandDefault{\capitalhungarumlaut}{\@tc@check@accent{\H}2\capitalhungarumlaut}
169 \DeclareTextCommandDefault{\capitalmacron}{\@tc@check@accent{\=2}\capitalmacron}
170 \DeclareTextCommandDefault{\capitalogonek}{\@tc@check@accent{\k}2\capitalogonek}
171 \DeclareTextCommandDefault{\capitalring}{\@tc@check@accent{\r}2\capitalring}
172 \DeclareTextCommandDefault{\capitaltie}{\@tc@check@accent{\t}2\capitaltie}
173 \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).

```

188 \DeclareTextCommandDefault{\newtie}{\@tc@check@accent{\t}2\newtie}
189 \DeclareTextCommandDefault{\capitalnewtie}{\@tc@check@accent{\t}2\capitalnewtie}
190
191

```

In Unicode engines we just execute the simple accents:

```

192 \else
193 \DeclareTextCommandDefault{\capitalacute}{\@tabacckludge'}
194 \DeclareTextCommandDefault{\capitalbreve}{\u}
195 \DeclareTextCommandDefault{\capitalcaron}{\v}
196 \DeclareTextCommandDefault{\capitalcedilla}{\c}
197 \DeclareTextCommandDefault{\capitalcircumflex}{\^}
198 \DeclareTextCommandDefault{\capitaldieresis}{"}
199 \DeclareTextCommandDefault{\capitaldotaccent}{\.}
200 \DeclareTextCommandDefault{\capitalgrave}{\@tabacckludge'}
201 \DeclareTextCommandDefault{\capitalhungarumlaut}{\H}
202 \DeclareTextCommandDefault{\capitalmacron}{\@tabacckludge=}
203 \DeclareTextCommandDefault{\capitalnewtie}{\t}
204 \DeclareTextCommandDefault{\capitalogonek}{\k}
205 \DeclareTextCommandDefault{\capitalring}{\r}
206 \DeclareTextCommandDefault{\capitaltie}{\t}
207 \DeclareTextCommandDefault{\capitaltilde}{\~}
208 \DeclareTextCommandDefault{\newtie}{\t}

```

```
209 \fi
```

The next two symbols exist in some fonts (faked?), but we ignore that to keep the subsets reasonable compact and most important linear.

```
210 \DeclareTextCommandDefault{\textlbrackdbl}{\tc@check@symbol2\textlbrackdbl}
211 \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.

```
214 \DeclareTextCommandDefault{\texteightoldstyle}{\tc@check@symbol2\texteightoldstyle}
215 \DeclareTextCommandDefault{\textfiveoldstyle}{\tc@check@symbol2\textfiveoldstyle}
216 \DeclareTextCommandDefault{\textfouroldstyle}{\tc@check@symbol2\textfouroldstyle}
217 \DeclareTextCommandDefault{\textnineoldstyle}{\tc@check@symbol2\textnineoldstyle}
218 \DeclareTextCommandDefault{\textoneoldstyle}{\tc@check@symbol2\textoneoldstyle}
219 \DeclareTextCommandDefault{\textsevenoldstyle}{\tc@check@symbol2\textsevenoldstyle}
220 \DeclareTextCommandDefault{\textsixoldstyle}{\tc@check@symbol2\textsixoldstyle}
221 \DeclareTextCommandDefault{\textthreeoldstyle}{\tc@check@symbol2\textthreeoldstyle}
222 \DeclareTextCommandDefault{\texttwooldstyle}{\tc@check@symbol2\texttwooldstyle}
223 \DeclareTextCommandDefault{\textzerooldstyle}{\tc@check@symbol2\textzerooldstyle}
```

The next set of glyphs is special to  $\text{T}_{\text{E}}\text{X}$  fonts (and available with a few older PS fonts supported through 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.

```
234 \DeclareTextCommandDefault{\textacutedbl}{\tc@check@symbol2\textacutedbl}
235 \DeclareTextCommandDefault{\textasciacute}{\tc@check@symbol2\textasciacute}
236 \DeclareTextCommandDefault{\textasciibreve}{\tc@check@symbol2\textasciibreve}
237 \DeclareTextCommandDefault{\textasciicaron}{\tc@check@symbol2\textasciicaron}
238 \DeclareTextCommandDefault{\textasciidieresis}{\tc@check@symbol2\textasciidieresis}
239 \DeclareTextCommandDefault{\textasciigrave}{\tc@check@symbol2\textasciigrave}
240 \DeclareTextCommandDefault{\textasciimacron}{\tc@check@symbol2\textasciimacron}
241 \DeclareTextCommandDefault{\textgravedbl}{\tc@check@symbol2\textgravedbl}
242 \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  $\text{T}_{\text{E}}\text{X}$  world.

```

252 \DeclareTextCommandDefault{\textbaht}
253 {\tc@check@symbol2{textbaht}}
254 \DeclareTextCommandDefault{\textbigcircle}
255 {\tc@check@symbol2{textbigcircle}}
256 \DeclareTextCommandDefault{\textborn}
257 {\tc@check@symbol2{textborn}}
258 \DeclareTextCommandDefault{\textcentoldstyle}
259 {\tc@check@symbol2{textcentoldstyle}}
260 \DeclareTextCommandDefault{\textcircledP}
261 {\tc@check@symbol2{textcircledP}}
262 \DeclareTextCommandDefault{\textcopyleft}
263 {\tc@check@symbol2{textcopyleft}}
264 \DeclareTextCommandDefault{\textdblhyphenchar}
265 {\tc@check@symbol2{textdblhyphenchar}}
266 \DeclareTextCommandDefault{\textdblhyphen}
267 {\tc@check@symbol2{textdblhyphen}}
268 \DeclareTextCommandDefault{\textdied}
269 {\tc@check@symbol2{textdied}}
270 \DeclareTextCommandDefault{\textdiscount}
271 {\tc@check@symbol2{textdiscount}}
272 \DeclareTextCommandDefault{\textdivorced}
273 {\tc@check@symbol2{textdivorced}}
274 \DeclareTextCommandDefault{\textdollaroldstyle}
275 {\tc@check@symbol2{textdollaroldstyle}}
276 \DeclareTextCommandDefault{\textguarani}
277 {\tc@check@symbol2{textguarani}}
278 \DeclareTextCommandDefault{\textleaf}
279 {\tc@check@symbol2{textleaf}}
280 \DeclareTextCommandDefault{\textlquill}
281 {\tc@check@symbol2{textlquill}}
282 \DeclareTextCommandDefault{\textmarried}
283 {\tc@check@symbol2{textmarried}}
284 \DeclareTextCommandDefault{\textmho}
285 {\tc@check@symbol2{textmho}}
286 \DeclareTextCommandDefault{\textmusicalnote}
287 {\tc@check@symbol2{textmusicalnote}}
288 \DeclareTextCommandDefault{\textnaira}
289 {\tc@check@symbol2{textnaira}}
290 \DeclareTextCommandDefault{\textopenbullet}
291 {\tc@check@symbol2{textopenbullet}}
292 \DeclareTextCommandDefault{\textpeso}
293 {\tc@check@symbol2{textpeso}}
294 \DeclareTextCommandDefault{\textpilcrow}
295 {\tc@check@symbol2{textpilcrow}}
296 \DeclareTextCommandDefault{\textrecipe}
297 {\tc@check@symbol2{textrecipe}}
298 \DeclareTextCommandDefault{\textreferencemark}
299 {\tc@check@symbol2{textreferencemark}}
300 \DeclareTextCommandDefault{\textrquill}
301 {\tc@check@symbol2{textrquill}}
302 \DeclareTextCommandDefault{\textservicemark}
303 {\tc@check@symbol2{textservicemark}}
304 \DeclareTextCommandDefault{\textsurd}
305 {\tc@check@symbol2{textsurd}}

```

The `\textpertenthousand` also belongs in this group but here we have a choice: in T1 there is a 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.

```
306 \DeclareTextCommandDefault{\textpertenthousand}
307 {\tc@check@symbol2\textpertenthousand}
308 \UndeclareTextCommand{\textpertenthousand}{T1}
```

### 1.3 Unavailable in sub-encoding 3 and higher

Sub-encoding 2 is the one where we loose many symbols. In the higher-numbered sub-encodings we see only a few dropped additionally.

```
309 \DeclareTextCommandDefault{\textlangle}
310 {\tc@check@symbol3\textlangle}
311 \DeclareTextCommandDefault{\textrangle}
312 {\tc@check@symbol3\textrangle}
```

### 1.4 Unavailable in sub-encoding 4 and higher

```
313 \DeclareTextCommandDefault{\textcolonmonetary}
314 {\tc@check@symbol4\textcolonmonetary}
315 \DeclareTextCommandDefault{\textdong}
316 {\tc@check@symbol4\textdong}
317 \DeclareTextCommandDefault{\textdownarrow}
318 {\tc@check@symbol4\textdownarrow}
319 \DeclareTextCommandDefault{\textleftarrow}
320 {\tc@check@symbol4\textleftarrow}
321 \DeclareTextCommandDefault{\textlira}
322 {\tc@check@symbol4\textlira}
323 \DeclareTextCommandDefault{\textrightarrow}
324 {\tc@check@symbol4\textrightarrow}
325 \DeclareTextCommandDefault{\textuparrow}
326 {\tc@check@symbol4\textuparrow}
327 \DeclareTextCommandDefault{\textwon}
328 {\tc@check@symbol4\textwon}
```

### 1.5 Unavailable in sub-encoding 5 (most older PS fonts) and higher

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 glyphs that should be available in sub-encoding 5. Instead they show little squares, i.e., they produce "tofu" if you are unlucky.

But the coverage is so random that it is impossible to sort them properly and if we tried to ensure that they only typeset those glyphs that are really always available, we would have to put them all into sub-encoding 9; so putting them into 5 is really a compromise.

Modern fonts usually don't typeset a tofu character if a glyph is missing. They are therefore only classified as sub-encoding 5 if they really support its glyph set completely.

```

329 \DeclareTextCommandDefault{\textestimated}
330 {\text@check@symbol5\textestimated}
331 \DeclareTextCommandDefault{\textnumero}
332 {\text@check@symbol5\textnumero}

```

## 1.6 Unavailable in sub-encoding 6 and higher

```

333 \DeclareTextCommandDefault{\textflorin}
334 {\text@check@symbol6\textflorin}
335 \DeclareTextCommandDefault{\textcurrency}
336 {\text@check@symbol6\textcurrency}

```

## 1.7 Unavailable in sub-encoding 7 and higher

```

337 \DeclareTextCommandDefault{\textfractionsolidus}
338 {\text@check@symbol7\textfractionsolidus}
339 \DeclareTextCommandDefault{\textohm}
340 {\text@check@symbol7\textohm}
341 \DeclareTextCommandDefault{\textmu}
342 {\text@check@symbol7\textmu}
343 \DeclareTextCommandDefault{\textminus}
344 {\text@check@symbol7\textminus}

```

## 1.8 Unavailable in sub-encoding 8 and higher

```

345 \DeclareTextCommandDefault{\textblank}
346 {\text@check@symbol8\textblank}
347 \DeclareTextCommandDefault{\textinterrobangdown}
348 {\text@check@symbol8\textinterrobangdown}
349 \DeclareTextCommandDefault{\textinterrobang}
350 {\text@check@symbol8\textinterrobang}

```

Fonts with this sub-encoding don't have a Euro symbol, but instead of substituting we fake it.

```

351 \DeclareTextCommandDefault{\texteuro}
352 {\CheckEncodingSubset\UseTextSymbol{TS1}\text@fake@euro{8}\texteuro}

```

## 1.9 Unavailable in Sub-encoding 9 (most missing)

```

353 \DeclareTextCommandDefault{\textcelsius}
354 {\text@check@symbol9\textcelsius}
355 \DeclareTextCommandDefault{\textonesuperior}
356 {\text@check@symbol9\textonesuperior}
357 \DeclareTextCommandDefault{\textthreequartersemdash}
358 {\text@check@symbol9\textthreequartersemdash}
359 \DeclareTextCommandDefault{\textthreesuperior}
360 {\text@check@symbol9\textthreesuperior}
361 \DeclareTextCommandDefault{\texttwelveudash}
362 {\text@check@symbol9\texttwelveudash}
363 \DeclareTextCommandDefault{\texttwosuperior}
364 {\text@check@symbol9\texttwosuperior}
365 \DeclareTextCommandDefault{\textbardbl}
366 {\text@check@symbol9\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).

```
367 \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.

```
368 \%DeclarerTextSymbol{\textcopyleft}{TS1}{171}
369 \%DeclarerTextSymbol{\textdblhyphen}{TS1}{45}
370 \%DeclarerTextSymbol{\textdblhyphenchar}{TS1}{127}
371 \%DeclarerTextSymbol{\textquotestraightbase}{TS1}{13}
372 \%DeclarerTextSymbol{\textquotestraightdblbase}{TS1}{18}
373 \%DeclarerTextSymbol{\textleaf}{TS1}{108}
374 \%DeclarerTextSymbol{\texttwelveudash}{TS1}{21}
375 \%DeclarerTextSymbol{\textthreequartersdash}{TS1}{22}
```

If oldstyle numerals are asked for we just use \oldstylenums.

```
376 \DeclarerTextCommand{\textzerooldstyle} \UnicodeEncodingName{\oldstylenums{0}}
377 \DeclarerTextCommand{\textoneoldstyle} \UnicodeEncodingName{\oldstylenums{1}}
378 \DeclarerTextCommand{\texttwooldstyle} \UnicodeEncodingName{\oldstylenums{2}}
379 \DeclarerTextCommand{\textthreeoldstyle} \UnicodeEncodingName{\oldstylenums{3}}
380 \DeclarerTextCommand{\textfouroldstyle} \UnicodeEncodingName{\oldstylenums{4}}
381 \DeclarerTextCommand{\textfiveoldstyle} \UnicodeEncodingName{\oldstylenums{5}}
382 \DeclarerTextCommand{\textsixoldstyle} \UnicodeEncodingName{\oldstylenums{6}}
383 \DeclarerTextCommand{\textsevenoldstyle} \UnicodeEncodingName{\oldstylenums{7}}
384 \DeclarerTextCommand{\texteightoldstyle} \UnicodeEncodingName{\oldstylenums{8}}
385 \DeclarerTextCommand{\textnineoldstyle} \UnicodeEncodingName{\oldstylenums{9}}
```

These have Unicode slots so this should be integrated into TU explicitly

```
386 \DeclarerTextSymbol{\textpilcrow} \UnicodeEncodingName{"00B6}
387 \DeclarerTextSymbol{\textborn} \UnicodeEncodingName{"002A}
388 \DeclarerTextSymbol{\textdied} \UnicodeEncodingName{"2020}
389 \DeclarerTextSymbol{\textlbrackdbl} \UnicodeEncodingName{"27E6}
390 \DeclarerTextSymbol{\textrbrackdbl} \UnicodeEncodingName{"27E7}
391 \DeclarerTextSymbol{\textguarani} \UnicodeEncodingName{"20B2}
```

We could make \textcentoldstyle and \textdollaroldstyle point to dollar and cent in the Unicode encoding

```
392 \%DeclarerTextSymbol{\textcentoldstyle} \UnicodeEncodingName{"00A2}
393 \%DeclarerTextSymbol{\textdollaroldstyle} \UnicodeEncodingName{"0024}
```

but I think it is better to pick them up from TS1 even if that usually means LMR fonts

```
394 \DeclarerTextSymbol{\textdollaroldstyle}{TS1}{138}
395 \DeclarerTextSymbol{\textcentoldstyle} {TS1}{139}
396 \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 T<sub>E</sub>X 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.

```
397 \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.

```
398 \DeclareEncodingSubset{TS1}{ccr} {0}
399 \DeclareEncodingSubset{TS1}{cmbr} {0}
400 \DeclareEncodingSubset{TS1}{cmr} {0}
401 \DeclareEncodingSubset{TS1}{cmss} {0}
402 \DeclareEncodingSubset{TS1}{cmtl} {0}
403 \DeclareEncodingSubset{TS1}{cmtt} {0}
404 \DeclareEncodingSubset{TS1}{cmvtt} {0}
405 \DeclareEncodingSubset{TS1}{pxr} {0}
406 \DeclareEncodingSubset{TS1}{pxss} {0}
407 \DeclareEncodingSubset{TS1}{pxtt} {0}
408 \DeclareEncodingSubset{TS1}{qag} {0}
409 \DeclareEncodingSubset{TS1}{qbk} {0}
410 \DeclareEncodingSubset{TS1}{qcr} {0}
411 \DeclareEncodingSubset{TS1}{qcs} {0}
412 \DeclareEncodingSubset{TS1}{qhvc} {0}
413 \DeclareEncodingSubset{TS1}{qhv} {0}
414 \DeclareEncodingSubset{TS1}{qpl} {0}
415 \DeclareEncodingSubset{TS1}{qtm} {0}
416 \DeclareEncodingSubset{TS1}{qzc} {0}
417 \DeclareEncodingSubset{TS1}{txr} {0}
418 \DeclareEncodingSubset{TS1}{txss} {0}
419 \DeclareEncodingSubset{TS1}{txtt} {0}

420 \DeclareEncodingSubset{TS1}{lmr} {1}
421 \DeclareEncodingSubset{TS1}{lmdh} {1}
422 \DeclareEncodingSubset{TS1}{lmss} {1}
423 \DeclareEncodingSubset{TS1}{lmssq} {1}
424 \DeclareEncodingSubset{TS1}{lmvtt} {1}
```

The lmtt family is missing TM, SM, and perthousand for some reason, so the first safe sub-encoding would be 2, but that is then missing out a huge number of glyphs that are available, so we claim it is sub-encoding 1 even if this can lead to missing glyphs.

```
425 \DeclareEncodingSubset{TS1}{lmtt} {1} % missing TM, SM and perthousand
```

The next three families have been removed from TeX Live, but we keep the definitions

```
426 \DeclareEncodingSubset{TS1}{ptmx} {2}
427 \DeclareEncodingSubset{TS1}{ptmj} {2}
428 \DeclareEncodingSubset{TS1}{u18} {2}
```

The next set are the early PostScript font implementations, these days there are better alternatives, but .... Note that, their virtual fonts contain a lot of “tofu” in form of black squares, thus they don’t even give a missing character warning if you select such a glyph. This is why they are set as sub-encoding 5.

```
429 \DeclareEncodingSubset{TS1}{bch} {5} % tofu for blank, ohm
430 \DeclareEncodingSubset{TS1}{futj} {5} % tofu for blank, interrobang/down, ohm
431 \DeclareEncodingSubset{TS1}{futs} {5} % tofu for blank, ohm
432 \DeclareEncodingSubset{TS1}{futx} {5} % probably (currently broken distrib)
433 \DeclareEncodingSubset{TS1}{pag} {5} % tofu for blank, interrobang/down, ohm
434 \DeclareEncodingSubset{TS1}{pbk} {5} % tofu for blank, interrobang/down, ohm
```

```

435 \DeclareEncodingSubset{TS1}{pcr} {5} % tofu for blank, interrobang/down, ohm
436 \DeclareEncodingSubset{TS1}{phv} {5} % tofu for blank, interrobang/down, ohm
437 \DeclareEncodingSubset{TS1}{pnc} {5} % tofu for blank, interrobang/down, ohm
438 \DeclareEncodingSubset{TS1}{pplj}{5} % tofu for blank
439 \DeclareEncodingSubset{TS1}{pplx}{5} % tofu for blank
440 \DeclareEncodingSubset{TS1}{ppl} {5} % tofu for blank interrobang/down
441 \DeclareEncodingSubset{TS1}{ptm} {5} % tofu for blank, interrobang/down, ohm
442 \DeclareEncodingSubset{TS1}{pzc} {5} % tofu for blank, interrobang/down, ohm
443 \DeclareEncodingSubset{TS1}{ul9} {5} % tofu for blank, interrobang/down, ohm

The next set suffers from the same problem and they contain even fewer real glyphs.

444 \DeclareEncodingSubset{TS1}{dayroms}{6} % tofu for blank, interrobang/down, ohm
445 \DeclareEncodingSubset{TS1}{dayrom} {6} % tofu for blank, interrobang/down, ohm
446 \DeclareEncodingSubset{TS1}{augie}{8} % really only missing euro
447 \DeclareEncodingSubset{TS1}{put} {8}
448 \DeclareEncodingSubset{TS1}{uag} {8} % probably (currently broken distrib)
449 \DeclareEncodingSubset{TS1}{ugq} {8}

450 \DeclareEncodingSubset{TS1}{zi4} {9}

```

LucidaBright (sold through TUG) probably not quite correct, I guess as I have the older fonts ...

```

451 \DeclareEncodingSubset{TS1}{hls} {5}
452 \DeclareEncodingSubset{TS1}{hlst} {5}
453 \DeclareEncodingSubset{TS1}{hlct} {5}
454 \DeclareEncodingSubset{TS1}{hh} {5}
455 \DeclareEncodingSubset{TS1}{hlx} {8}
456 \DeclareEncodingSubset{TS1}{hlce} {8}
457 \DeclareEncodingSubset{TS1}{hlcn} {8}
458 \DeclareEncodingSubset{TS1}{hlcw} {8}
459 \DeclareEncodingSubset{TS1}{hlcf} {8}

```

Below are the newer fonts that have support files for L<sup>A</sup>T<sub>E</sub>X. 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).

Encoding declarations for these font families shouldn’t really be in the kernel, but part of the .fd files for the family. When we introduced the concept in 2021 we had some hope that this would happen over time and that we could take the declarations out—after all it is nearly impossible to maintain it correctly in the kernel, given that fonts may get new glyphs added (happened for several of them in the recent year) which is something we wouldn’t notice. However, so far this hasn’t happened, so in 2024, I went through the current set and adjusted the declarations in several places.

Next four are wrong and still need adjustment:

```

460 \DeclareEncodingSubset{TS1}{lato-*} {0} % with a bunch of tofu inside
461 \DeclareEncodingSubset{TS1}{opensans-*} {0} % with a bunch of tofu inside
462 \DeclareEncodingSubset{TS1}{cantarell-*} {0} % with a bunch of tofu inside
463 \DeclareEncodingSubset{TS1}{tli} {1} % with lots of tofu inside
464 \DeclareEncodingSubset{TS1}{fbb-*} {2} % missing centoldstyle

```

```

465 \DeclareEncodingSubset{TS1}{Alegreya-*} {2}
466 \DeclareEncodingSubset{TS1}{AlegreyaSans-*} {2}
467 \DeclareEncodingSubset{TS1}{BaskervilleF-*} {2}
468 \DeclareEncodingSubset{TS1}{DejaVuSans-TLF} {2}
469 \DeclareEncodingSubset{TS1}{DejaVuSansCondensed-TLF} {2}

```

Next one is missing \textfractionsolidus but is otherwise completely sub-encoding 2 so we use that sub-encoding.

```

470 \DeclareEncodingSubset{TS1}{DejaVuSansMono-TLF} {2}
471 \DeclareEncodingSubset{TS1}{EBGaramond-*} {2}
472 \DeclareEncodingSubset{TS1}{Merriwthr-OsF} {2}
473 \DeclareEncodingSubset{TS1}{MerriwthrSans-OsF} {2}
474 \DeclareEncodingSubset{TS1}{Montserrat-*} {2}
475 \DeclareEncodingSubset{TS1}{MontserratAlternates-*} {2}
476 \DeclareEncodingSubset{TS1}{NotoSansMono-TLF} {2}
477 \DeclareEncodingSubset{TS1}{NotoSansMono-T0sF} {2}
478 \DeclareEncodingSubset{TS1}{Tempora-TLF} {2}
479 \DeclareEncodingSubset{TS1}{Tempora-T0sF} {2}
480 \DeclareEncodingSubset{TS1}{XCharter-TLF} {2}
481 \DeclareEncodingSubset{TS1}{XCharter-T0sF} {2}
482 \DeclareEncodingSubset{TS1}{Ferewhon-*} {2}
483 \DeclareEncodingSubset{TS1}{Arimo-TLF} {3}
484 \DeclareEncodingSubset{TS1}{Crlt-*} {3}
485 \DeclareEncodingSubset{TS1}{IBMPlexMono-TLF} {3}
486 \DeclareEncodingSubset{TS1}{IBMPlexSans-TLF} {3}
487 \DeclareEncodingSubset{TS1}{IBMPlexSerif-TLF} {3}
488 \DeclareEncodingSubset{TS1}{SourceCodePro-TLF} {3}
489 \DeclareEncodingSubset{TS1}{SourceCodePro-T0sF} {3}
490 \DeclareEncodingSubset{TS1}{SourceSansPro-*} {3}
491 \DeclareEncodingSubset{TS1}{SourceSerifPro-*} {3}
492 \DeclareEncodingSubset{TS1}{Tinos-TLF} {3}
493 \DeclareEncodingSubset{TS1}{AccanthisADFStdNoThree-LF}{4}
494 \DeclareEncodingSubset{TS1}{Cabin-TLF} {4}
495 \DeclareEncodingSubset{TS1}{Caladea-TLF} {4}
496 \DeclareEncodingSubset{TS1}{Chivo-*} {4}
497 \DeclareEncodingSubset{TS1}{ClearSans-TLF} {4}
498 \DeclareEncodingSubset{TS1}{Coelacanth-LF} {4}
499 \DeclareEncodingSubset{TS1}{CrimsonPro-*} {4}
500 \DeclareEncodingSubset{TS1}{FiraMono-TLF} {4}
501 \DeclareEncodingSubset{TS1}{FiraMono-T0sF} {4}
502 \DeclareEncodingSubset{TS1}{FiraSans-*} {4}
503 \DeclareEncodingSubset{TS1}{Go-TLF} {4}
504 \DeclareEncodingSubset{TS1}{GoMono-TLF} {4}
505 \DeclareEncodingSubset{TS1}{InriaSans-*} {4}
506 \DeclareEncodingSubset{TS1}{InriaSerif-*} {4}
507 \DeclareEncodingSubset{TS1}{LibertinusSans-*} {4}
508 \DeclareEncodingSubset{TS1}{LibertinusSerif-*} {4}
509 \DeclareEncodingSubset{TS1}{LibreBodoni-TLF} {4}
510 \DeclareEncodingSubset{TS1}{LibreFranklin-TLF} {4}
511 \DeclareEncodingSubset{TS1}{LinguisticsPro-LF} {4}
512 \DeclareEncodingSubset{TS1}{LinguisticsPro-OsF} {4}
513 \DeclareEncodingSubset{TS1}{LinuxBiolinumT-*} {4}
514 \DeclareEncodingSubset{TS1}{LinuxLibertineT-*} {4}

```

```

515 \DeclareEncodingSubset{TS1}{MintSpirit-*} {4}
516 \DeclareEncodingSubset{TS1}{MintSpiritNoTwo-*} {4}
517 \DeclareEncodingSubset{TS1}{PTMono-TLF} {4}
518 \DeclareEncodingSubset{TS1}{PTSans-TLF} {4}
519 \DeclareEncodingSubset{TS1}{PTSansCaption-TLF} {4}
520 \DeclareEncodingSubset{TS1}{PTSansNarrow-TLF} {4}
521 \DeclareEncodingSubset{TS1}{PTSerif-TLF} {4}
522 \DeclareEncodingSubset{TS1}{PTSerifCaption-TLF} {4}
523 \DeclareEncodingSubset{TS1}{Raleway-TLF} {4}
524 \DeclareEncodingSubset{TS1}{Raleway-T0sF} {4}
525 \DeclareEncodingSubset{TS1}{Roboto-*} {4}
526 \DeclareEncodingSubset{TS1}{RobotoMono-TLF} {4}
527 \DeclareEncodingSubset{TS1}{RobotoSlab-TLF} {4}
528 \DeclareEncodingSubset{TS1}{Rosario-*} {4}
529 \DeclareEncodingSubset{TS1}{SticksTooText-*} {4}
530 \DeclareEncodingSubset{TS1}{UniversalisADFStd-LF} {4}

531 \DeclareEncodingSubset{TS1}{Almnndr-OsF} {5}
532 \DeclareEncodingSubset{TS1}{Baskervaldx-*} {5}
533 \DeclareEncodingSubset{TS1}{Bttr-TLF} {5}
534 \DeclareEncodingSubset{TS1}{Cinzel-LF} {5}
535 \DeclareEncodingSubset{TS1}{CinzelDecorative-LF} {5}
536 \DeclareEncodingSubset{TS1}{Cochineal-*} {5}
537 \DeclareEncodingSubset{TS1}{DejaVuSerif-TLF} {5}
538 \DeclareEncodingSubset{TS1}{DejaVuSerifCondensed-TLF} {5}
539 \DeclareEncodingSubset{TS1}{GilliusADF-LF} {5}
540 \DeclareEncodingSubset{TS1}{GilliusADFCond-LF} {5}
541 \DeclareEncodingSubset{TS1}{GilliusADFNNoTwo-LF} {5}
542 \DeclareEncodingSubset{TS1}{GilliusADFNNoTwoCond-LF} {5}
543 \DeclareEncodingSubset{TS1}{OldStandard-TLF} {5}
544 \DeclareEncodingSubset{TS1}{PlyfrDisplay-TLF} {5}
545 \DeclareEncodingSubset{TS1}{PlyfrDisplay-T0sF} {5}
546 \DeclareEncodingSubset{TS1}{TheanoDidot-TLF} {5}
547 \DeclareEncodingSubset{TS1}{TheanoDidot-T0sF} {5}
548 \DeclareEncodingSubset{TS1}{TheanoModern-TLF} {5}
549 \DeclareEncodingSubset{TS1}{TheanoModern-T0sF} {5}
550 \DeclareEncodingSubset{TS1}{TheanoOldStyle-TLF} {5}
551 \DeclareEncodingSubset{TS1}{TheanoOldStyle-T0sF} {5}
552 \DeclareEncodingSubset{TS1}{charssil-TLF} {5}

553 \DeclareEncodingSubset{TS1}{Crimson-TLF} {6}
554 \DeclareEncodingSubset{TS1}{LibertinusSerifDisplay-LF} {6}
555 \DeclareEncodingSubset{TS1}{LinuxLibertineDisplayT-*} {6}
556 \DeclareEncodingSubset{TS1}{LinuxLibertineMonOT-LF} {6}
557 \DeclareEncodingSubset{TS1}{LinuxLibertineMonOT-TLF} {6}
558 \DeclareEncodingSubset{TS1}{Ovrlck-LF} {6}

559 \DeclareEncodingSubset{TS1}{ComicNeue-TLF} {7}
560 \DeclareEncodingSubset{TS1}{ComicNeueAngular-TLF} {7}
561 \DeclareEncodingSubset{TS1}{CormorantGaramond-*} {7}
562 \DeclareEncodingSubset{TS1}{Heuristica-TLF} {7}
563 \DeclareEncodingSubset{TS1}{Heuristica-T0sF} {7}
564 \DeclareEncodingSubset{TS1}{IMFELLEnglish-TLF} {7}
565 \DeclareEncodingSubset{TS1}{LibreBskrvl-TLF} {7}
566 \DeclareEncodingSubset{TS1}{LibreCsln-*} {7}
567 \DeclareEncodingSubset{TS1}{Lbstr-LF} {7}

```

```

568 \DeclareEncodingSubset{TS1}{Mrcls-LF} {7}
Strangely enough NotoSerif and NotoSans are sub-encoding 7 as they are missing
\textminus and several other glyphs. In contrast the NotoSansMono is far more complete.

569 \DeclareEncodingSubset{TS1}{NotoSans-*} {7}
570 \DeclareEncodingSubset{TS1}{NotoSerif-*} {7}
571 \DeclareEncodingSubset{TS1}{Quattro-LF} {7}
572 \DeclareEncodingSubset{TS1}{QuattroSans-LF} {7}
573 \DeclareEncodingSubset{TS1}{Frm-LF} {7} % the superiors are missing
574 \DeclareEncodingSubset{TS1}{LibertinusMono-TLF} {8}
575 \DeclareEncodingSubset{TS1}{AlgolRevived-TLF} {9}

```

## 4 Legacy symbol support for lists and footnote symbols

```

\UseLegacyTextSymbols

576 \def\UseLegacyTextSymbols{%
577 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}%
578 \DeclareTextSymbolDefault{\textbardbl}{OMS}%
579 \DeclareTextSymbolDefault{\textbullet}{OMS}%
580 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}%
581 \DeclareTextSymbolDefault{\textdagger}{OMS}%
582 \DeclareTextSymbolDefault{\textparagraph}{OMS}%
583 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}%
584 \DeclareTextSymbolDefault{\textsection}{OMS}%
585 \UndeclareTextCommand{\textsection}{T1}%
586 \expandafter\let\csname oldstylenums \expandafter\endcsname
587 \csname legacyoldstylenums \endcsname
588 }

```

(End of definition for \UseLegacyTextSymbols.)

\textlegacyasteriskcentered Here are new names for the legacy symbols that L<sup>A</sup>T<sub>E</sub>X used to pick up from the OMS encoded fonts (and used for 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.

```

589 \DeclareTextSymbol{\textlegacyasteriskcentered}{OMS}{3} % "03
590 \DeclareTextSymbol{\textlegacybardbl}{OMS}{107} % "6B
591 \DeclareTextSymbol{\textlegacybullet}{OMS}{15} % "0F
592 \DeclareTextSymbol{\textlegacydaggerdbl}{OMS}{122} % "7A
593 \DeclareTextSymbol{\textlegacydagger}{OMS}{121} % "79
594 \DeclareTextSymbol{\textlegacyparagraph}{OMS}{123} % "7B
595 \DeclareTextSymbol{\textlegacyperiodcentered}{OMS}{1} % "01
596 \DeclareTextSymbol{\textlegacysection}{OMS}{120} % "78

```

```

597 \DeclareTextSymbolDefault{\textlegacyasteriskcentered}{OMS}
598 \DeclareTextSymbolDefault{\textlegacybardbl}{OMS}
599 \DeclareTextSymbolDefault{\textlegacybullet}{OMS}
600 \DeclareTextSymbolDefault{\textlegacydaggerdbl}{OMS}
601 \DeclareTextSymbolDefault{\textlegacydagger}{OMS}
602 \DeclareTextSymbolDefault{\textlegacyparagraph}{OMS}
603 \DeclareTextSymbolDefault{\textlegacyperiodcentered}{OMS}
604 \DeclareTextSymbolDefault{\textlegacysection}{OMS}

(End of definition for \textlegacyasteriskcentered and others.)

Supporting rollback ...

605 //ekernel | latexrelease)
606 <latexrelease>
607 <latexrelease>\IncludeInRelease{0000/00/00}%
608 <latexrelease> {ltxtextcomp}{Undefine text companion symbols}%
609 <latexrelease>
610 <latexrelease>\DeclareRobustCommand\oldstylenums[1]{%
611 <latexrelease> \begingroup
612 <latexrelease> \spaceskip\fontdimen\tw@\font
613 <latexrelease> \usefont{OML}{\rmdefault}{\f@series}{it}%
614 <latexrelease> \mathgroup\symletters #1%
615 <latexrelease> \endgroup
616 <latexrelease>}
617 <latexrelease>\let\legacyoldstylenums\@undefined
618 <latexrelease>\def\textracompsubstdefault{cmr}
619 <latexrelease>
620 <latexrelease>\let\DeclareEncodingSubset\@undefined
621 <latexrelease>\let\CheckEncodingSubset\@undefined
622 <latexrelease>
623 <latexrelease>\DeclareTextSymbolDefault{\textdollar}{OT1}
624 <latexrelease>\DeclareTextSymbolDefault{\textsterling}{OT1}
625 <latexrelease>\DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
626 <latexrelease> \ifdim \fontdimen\@ne\font >\z@
627 <latexrelease> \slshape
628 <latexrelease> \else
629 <latexrelease> \upshape
630 <latexrelease> \fi
631 <latexrelease> \char`\$\egroup}
632 <latexrelease>\DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
633 <latexrelease> \ifdim \fontdimen\@ne\font >\z@
634 <latexrelease> \itshape
635 <latexrelease> \else
636 <latexrelease> \fontshape{ui}\selectfont
637 <latexrelease> \fi
638 <latexrelease> \char`\$\egroup}
639 <latexrelease>\DeclareTextCommand{\textperthousand}{T1}
640 <latexrelease> {\%\char 24 }
641 <latexrelease>
642 <latexrelease>\DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
643 <latexrelease>\DeclareTextSymbolDefault{\textbullet}{OMS}
644 <latexrelease>\DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
645 <latexrelease>\DeclareTextSymbolDefault{\textdagger}{OMS}
646 <latexrelease>\DeclareTextSymbolDefault{\textparagraph}{OMS}
647 <latexrelease>\DeclareTextSymbolDefault{\textperiodcentered}{OMS}

```

```

648 〈latexrelease〉\DeclareTextSymbolDefault{\textsection}{OMS}
649 〈latexrelease〉
650 〈latexrelease〉\DeclareTextSymbolDefault{\textbardbl}{OMS}
651 〈latexrelease〉\let\textbrokenbar\@undefined
652 〈latexrelease〉\let\textcelsius\@undefined
653 〈latexrelease〉\let\textcent\@undefined
654 〈latexrelease〉\DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
655 〈latexrelease〉
656 〈latexrelease〉\let\textdegree\@undefined
657 〈latexrelease〉\let\textdiv\@undefined
658 〈latexrelease〉\let\textlnot\@undefined
659 〈latexrelease〉\let\textonehalf\@undefined
660 〈latexrelease〉\let\textonequarter\@undefined
661 〈latexrelease〉\let\textonesuperior\@undefined
662 〈latexrelease〉\DeclareTextCommandDefault{\textordfeminine}{a}
663 〈latexrelease〉
664 〈latexrelease〉\DeclareTextCommandDefault{\textordmasculine}{o}
665 〈latexrelease〉
666 〈latexrelease〉\let\textpm\@undefined
667 〈latexrelease〉\let\textquotesingle\@undefined
668 〈latexrelease〉\let\textquotestraightbase\@undefined
669 〈latexrelease〉\let\textquotestraightdblbase\@undefined
670 〈latexrelease〉\DeclareTextCommandDefault{\textregistered}{\textcircled{R}}
671 〈latexrelease〉
672 〈latexrelease〉\check@mathfonts\fontsize\sf@size\z@\math@fontsf@false\selectfont R\}
673 〈latexrelease〉
674 〈latexrelease〉\let\textthreequartersemdash\@undefined
675 〈latexrelease〉\let\textthreequarters\@undefined
676 〈latexrelease〉\let\textthreesuperior\@undefined
677 〈latexrelease〉\let\textttimes\@undefined
678 〈latexrelease〉\DeclareTextCommandDefault{\texttrademark}{TM}
679 〈latexrelease〉
680 〈latexrelease〉\let\texttwelveudash\@undefined
681 〈latexrelease〉\let\texttwosuperior\@undefined
682 〈latexrelease〉\let\textyen\@undefined
683 〈latexrelease〉
684 〈latexrelease〉\let\textcapitalcompwordmark\@undefined
685 〈latexrelease〉\let\textascendercompwordmark\@undefined
686 〈latexrelease〉
687 〈latexrelease〉\DeclareTextAccentDefault{\textcircled}{OMS}
688 〈latexrelease〉\DeclareTextAccentDefault{\t}{OML}
689 〈latexrelease〉
690 〈latexrelease〉\let\capitalacute\@undefined
691 〈latexrelease〉\let\capitalbreve\@undefined
692 〈latexrelease〉\let\capitalcaron\@undefined
693 〈latexrelease〉\let\capitalcedilla\@undefined
694 〈latexrelease〉\let\capitalcircumflex\@undefined
695 〈latexrelease〉\let\capitaldieresis\@undefined
696 〈latexrelease〉\let\capitaldotaccent\@undefined
697 〈latexrelease〉\let\capitalgrave\@undefined
698 〈latexrelease〉\let\capitalhungarumlaut\@undefined
699 〈latexrelease〉\let\capitalmacron\@undefined
700 〈latexrelease〉\let\capitalnewtie\@undefined
701 〈latexrelease〉\let\capitalogonek\@undefined

```

```

702 〈\latexrelease〉\let\capitalring\@undefined
703 〈\latexrelease〉\let\capitaltie\@undefined
704 〈\latexrelease〉\let\capitaltilde\@undefined
705 〈\latexrelease〉\let\newtie\@undefined
706 〈\latexrelease〉
707 〈\latexrelease〉\let\textlbrackdbl\@undefined
708 〈\latexrelease〉\let\textrbrackdbl\@undefined
709 〈\latexrelease〉
710 〈\latexrelease〉\let\texteightoldstyle\@undefined
711 〈\latexrelease〉\let\textfiveoldstyle\@undefined
712 〈\latexrelease〉\let\textfouroldstyle\@undefined
713 〈\latexrelease〉\let\textnineoldstyle\@undefined
714 〈\latexrelease〉\let\textoneoldstyle\@undefined
715 〈\latexrelease〉\let\textsevenoldstyle\@undefined
716 〈\latexrelease〉\let\textsixoldstyle\@undefined
717 〈\latexrelease〉\let\textthreeoldstyle\@undefined
718 〈\latexrelease〉\let\texttwooldstyle\@undefined
719 〈\latexrelease〉\let\textzerooldstyle\@undefined
720 〈\latexrelease〉
721 〈\latexrelease〉\let\textacutedbl\@undefined
722 〈\latexrelease〉\let\textasciacute\@undefined
723 〈\latexrelease〉\let\textasciibreve\@undefined
724 〈\latexrelease〉\let\textasciicaron\@undefined
725 〈\latexrelease〉\let\textasciidieresis\@undefined
726 〈\latexrelease〉\let\textasciigrave\@undefined
727 〈\latexrelease〉\let\textasciimacron\@undefined
728 〈\latexrelease〉\let\textgravedbl\@undefined
729 〈\latexrelease〉\let\texttildelow\@undefined
730 〈\latexrelease〉
731 〈\latexrelease〉\let\textbaht\@undefined
732 〈\latexrelease〉\let\textbigcircle\@undefined
733 〈\latexrelease〉\let\textborn\@undefined
734 〈\latexrelease〉\let\textcentoldstyle\@undefined
735 〈\latexrelease〉\let\textcircledP\@undefined
736 〈\latexrelease〉\let\textcopyleft\@undefined
737 〈\latexrelease〉\let\textdblyhyphenchar\@undefined
738 〈\latexrelease〉\let\textdblyhyphen\@undefined
739 〈\latexrelease〉\let\textdied\@undefined
740 〈\latexrelease〉\let\textdiscount\@undefined
741 〈\latexrelease〉\let\textdivorced\@undefined
742 〈\latexrelease〉\let\textdollaroldstyle\@undefined
743 〈\latexrelease〉\let\textguarani\@undefined
744 〈\latexrelease〉\let\textleaf\@undefined
745 〈\latexrelease〉\let\textlquill\@undefined
746 〈\latexrelease〉\let\textmarried\@undefined
747 〈\latexrelease〉\let\textmho\@undefined
748 〈\latexrelease〉\let\textmusicalnote\@undefined
749 〈\latexrelease〉\let\textnaira\@undefined
750 〈\latexrelease〉\let\textopenbullet\@undefined
751 〈\latexrelease〉\let\textpeso\@undefined
752 〈\latexrelease〉\let\textpilcrow\@undefined
753 〈\latexrelease〉\let\textrecipe\@undefined
754 〈\latexrelease〉\let\textreferencemark\@undefined
755 〈\latexrelease〉\let\textrquill\@undefined

```

```

756 〈\latexrelease〉\let\textservicemark\@undefined
757 〈\latexrelease〉\let\textsurd\@undefined
758 〈\latexrelease〉
759 〈\latexrelease〉\DeclareTextCommand{\textpertenthousand}{T1}
760 〈\latexrelease〉 {\%\char 24\char 24 }
761 〈\latexrelease〉
762 〈\latexrelease〉\let\textlangle\@undefined
763 〈\latexrelease〉\let\textrangle\@undefined
764 〈\latexrelease〉
765 〈\latexrelease〉\let\textcolonmonetary\@undefined
766 〈\latexrelease〉\let\textdong\@undefined
767 〈\latexrelease〉\let\textdownarrow\@undefined
768 〈\latexrelease〉\let\textleftarrow\@undefined
769 〈\latexrelease〉\let\textlira\@undefined
770 〈\latexrelease〉\let\textrightarrow\@undefined
771 〈\latexrelease〉\let\textuparrow\@undefined
772 〈\latexrelease〉\let\textwon\@undefined
773 〈\latexrelease〉
774 〈\latexrelease〉\let\textestimated\@undefined
775 〈\latexrelease〉\let\textnumero\@undefined
776 〈\latexrelease〉
777 〈\latexrelease〉\let\textflorin\@undefined
778 〈\latexrelease〉\let\textcurrency\@undefined
779 〈\latexrelease〉
780 〈\latexrelease〉\let\textfractionsolidus\@undefined
781 〈\latexrelease〉\let\textohm\@undefined
782 〈\latexrelease〉\let\textmu\@undefined
783 〈\latexrelease〉\let\textminus\@undefined
784 〈\latexrelease〉
785 〈\latexrelease〉\let\textblank\@undefined
786 〈\latexrelease〉\let\textinterrobangdown\@undefined
787 〈\latexrelease〉\let\textinterrobang\@undefined
788 〈\latexrelease〉
789 〈\latexrelease〉\let\texteuro\@undefined
790 〈\latexrelease〉
791 〈\latexrelease〉\let\textcelsius\@undefined
792 〈\latexrelease〉\let\textonesuperior\@undefined
793 〈\latexrelease〉\let\textthreequartersemdash\@undefined
794 〈\latexrelease〉\let\textthreesuperior\@undefined
795 〈\latexrelease〉\let\texttwelvedash\@undefined
796 〈\latexrelease〉\let\texttwosuperior\@undefined
797 〈\latexrelease〉\let\textbardbl\@undefined
798 〈\latexrelease〉
799 〈\latexrelease〉\let\UseLegacyTextSymbols\@undefined
800 〈\latexrelease〉\let\textlegacyasteriskcentered\@undefined
801 〈\latexrelease〉\let\textlegacybardbl\@undefined
802 〈\latexrelease〉\let\textlegacybullet\@undefined
803 〈\latexrelease〉\let\textlegacydaggerdbl\@undefined
804 〈\latexrelease〉\let\textlegacydagger\@undefined
805 〈\latexrelease〉\let\textlegacyparagraph\@undefined
806 〈\latexrelease〉\let\textlegacyperiodcentered\@undefined
807 〈\latexrelease〉\let\textlegacysection\@undefined
808 〈\latexrelease〉
809 〈\latexrelease〉\EndModuleRelease

```

## 5 The `textcomp` package

For any rollback request before 2018-08-11 we make an attempt by loading the 2018 version.

```
810 {*TS1sty}
811 \DeclareRelease{}{1997-12-01}{textcomp-2018-08-11.sty}
812 \DeclareRelease{}{2018-08-11}{textcomp-2018-08-11.sty}
813 \DeclareCurrentRelease{}{2020-02-02}
814
815 \ProvidesPackage{textcomp}
816 [2024/04/24 v2.1b Standard LaTeX package]
A precaution in case this is used without rebuilding the format.
817 \NeedsTeXFormat{LaTeX2e}[2020/02/02]

This is implemented by defining the default subset:
```

```
818 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
819 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
820 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{8}}
821 \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.

```
822 \DeclareOption{error}
823 {\gdef\tc@errorwarn{\PackageError{textcomp}}}
824 \DeclareOption{warn}
825 {\gdef\tc@errorwarn#1#2{\PackageWarning{textcomp}{#1}}}
826 \DeclareOption{info}
827 {\gdef\tc@errorwarn#1#2{\PackageInfo{textcomp}{#1}}}
828 \DeclareOption{quiet}{\gdef\tc@errorwarn#1#2{}}
```

The “force” option basically changes the sub-encoding to that of the default (which, unless changes, is 9 these days), i.e., it no longer depends on the font in use. This is mainly there because it might have been used in older documents, but not something that is recommended.

```
829 \DeclareOption{force}{%
830 \def\CheckEncodingSubset#1#2#3#4#5{%
831 \ifnum #4>%
832 0\csname #2:?\endcsname
833 \relax
834 \expandafter\@firstoftwo
835 \else
836 \expandafter\@secondoftwo
837 \fi
838 {#1{#2}}{#3}%
839 #5}%
840 }
841 \ProcessOptions\relax
```

There is not much else to do nowadays, because everything is already set up in the L<sup>A</sup>T<sub>E</sub>X kernel.

```
842 \InputIfFileExists{textcomp.cfg}
843 {\PackageInfo{textcomp}{Local configuration file used}}{}
844 </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 categorized in the sub-encodings only into 6 classes not 10.

```
845 <*TS1oldsty>
846 \ProvidesPackage{textcomp}
847 [2018/08/11 v2.0j Standard LATEX package]
```

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

Since many PostScript fonts only implement a subset of TS1 many commands only produce black blobs of ink. To resolve the resulting problems a number of options have been introduced and some code has been developed to distinguish sub-encodings.

The sub-encodings have a numerical id and are defined as follows for TS1:

#5 those TS1 symbols that are also in the ISO-Adobe character set; without `textcurrency`, which is often misused for the Euro. Older Type1 fonts from the non-T<sub>E</sub>X world provide only this subset.

#4 = #5 + `\texteuro`. Most newer fonts provide this.

#3 = #4 + `\textomega`. Can also be described as  $TS1 \cap (ISO\text{-}Adobe \cup MacRoman)$ . (Except for the missing "currency".)

#2 = #3 + `\textestimated` + `\textcurrency`. Can also be described as  $TS1 \cap Adobe\text{-}Western\text{-}2$ . 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:

```
848 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
849 \space\space 5 = only ISO-Adobe without
850 \string\textcurrency\MessageBreak
851 \space\space 4 = 5 + \string\texteuro\MessageBreak
852 \space\space 3 = 4 + \string\textohm\MessageBreak
853 \space\space 2 = 3 + \noexpand\textestimated+
854 \string\textcurrency\MessageBreak
855 \space\space 1 = TS1 - \noexpand\textcircled-
```

```

856 \string\t\MessageBreak
857 \space\space 0 = TS1 (full)\MessageBreak
858 Font families with sub-encoding setting implement\MessageBreak
859 only a restricted character set as indicated.\MessageBreak
860 Family '?' is the default used for unknown fonts.\MessageBreak
861 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., `cmtt`), and the subset encoding id as a third, (e.g., `0` for `cmtt`).

The default encoding subset to use when nothing is known about the current font family is named `?`.

```

862 \def\DeclareEncodingSubset#1#2#3{%
863 \@ifundefined{#1:#2}{%
864 {\@PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
865 {\@PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
866 \@namedef{#1:#2}{#3}%
867 }@\onlypreamble\DeclareEncodingSubset

```

(End of definition for `\DeclareEncodingSubset`.)

The options for the package are the following:

**safe** for unknown font families enables only symbols that are also in the ISO-Adobe character set; without "currency", which is often misused for the Euro. Older Type1 fonts from the non-TeX world provide only this subset.

**euro** enables the "safe" symbols plus the `\texteuro` command. Most newer fonts provide this.

**full** enables all `TS1` commands; useful only with fonts like EC or CM bright.

**almostfull** same as "full", except that `\textcircled` and `\t` are *not* redefined from their defaults to avoid that commands like `\copyright` suddenly no longer work.

**force** ignore all subset encoding definitions stored in the package itself or in the configuration file and always use the default subset as specified by one of the other options (seldom useful, only dangerous).

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

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

(End of definition for `\iftc@forced`.)

This is implemented by defining the default subset:

```

869 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
870 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
871 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
872 \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.

```
873 \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.

```

874 \def\tc@errorwarn{\PackageError}
875 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
876 \DeclareOption{quiet}{\gdef\tc@errorwarn#1#2#3{}}
877 \ExecuteOptions{almostfull}
878 \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.

```
879 \iftc@forced
```

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

```

880 \def\CheckEncodingSubset#1#2#3#4#5{%
881 \ifnum #4>%
882 0\csname #2:?\endcsname
883 \relax
884 \expandafter\@firstoftwo
885 \else
886 \expandafter\@secondoftwo
887 \fi
888 {#1{#2}}{#3}%
889 #5%
890 }
```

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.

```

891 \else
892 \def\CheckEncodingSubset#1#2#3#4#5{%
893 \ifnum #4>%
894 \expandafter\fix\csname #2:\f@family\endcsname\relax
895 0\csname #2:?\endcsname
896 \else
897 \csname #2:\f@family\endcsname
898 \fi
899 \relax
900 \expandafter\@firstoftwo
901 \else
902 \expandafter\@secondoftwo
903 \fi
```

```

904 {#1{#2}}{#3}%
905 #5%
906 }
907 \fi

```

(End of definition for \CheckEncodingSubset.)

\tc@subst

```

908 \def\tc@subst#1{%
909 \tc@errorwarn{textcomp}%
910 {Symbol \string#1 not provided by\MessageBreak
911 font family \f@family\space
912 in TS1 encoding.\MessageBreak Default family used instead}\@eha
913 \bgroup\fontfamily{textcomp}\substitution\selectfont#1\egroup
914 }

```

(End of definition for \tc@subst.)

\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.

```

915 % error commands take argument:
916 % #1 symbol to be used
917 \def\tc@error#1{%
918 \PackageError{textcomp}{% should be latex error if general
919 {Accent \string#1 not provided by\MessageBreak
920 font family \f@family\space
921 in TS1 encoding}\@eha
922 }

```

(End of definition for \tc@error.)

\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 “=“.

```

923 \def\tc@fake@euro#1{%
924 \leavevmode
925 \PackageInfo{textcomp}{Faking \noexpand#1 for font family
926 \f@family\MessageBreak in TS1 encoding}%
927 \valign{##\cr
928 \vfil\hbox to 0.07em{\dimen@\f@size\p@
929 \math@fontsfalse
930 \fontsize{.7\dimen@}\z@\selectfont=\hss}%
931 \vfil\cr%
932 \hbox{C}\crcr
933 }%
934 }

```

(End of definition for \tc@fake@euro.)

\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 `\textcurrency` is not available.

```
935 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
936 \def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
```

*(End of definition for `\tc@check@symbol` and `\tc@check@accent`.)*

We start with the commands that are “safe” and which can be unconditionally set up, first the accents...

```
937 \DeclareTextAccentDefault{\capitalcedilla}{TS1}
938 \DeclareTextAccentDefault{\capitalogonek}{TS1}
939 \DeclareTextAccentDefault{\capitalgrave}{TS1}
940 \DeclareTextAccentDefault{\capitalacute}{TS1}
941 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}
942 \DeclareTextAccentDefault{\capitaltilde}{TS1}
943 \DeclareTextAccentDefault{\capitaldieresis}{TS1}
944 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
945 \DeclareTextAccentDefault{\capitalring}{TS1}
946 \DeclareTextAccentDefault{\capitalcaron}{TS1}
947 \DeclareTextAccentDefault{\capitalbreve}{TS1}
948 \DeclareTextAccentDefault{\capitalmacron}{TS1}
949 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}
```

...and then the other glyphs.

```
950 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
951 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
952 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
953 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
954 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}
955 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
956 \DeclareTextSymbolDefault{\textdollar}{TS1}
957 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
958 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
959 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
960 \DeclareTextSymbolDefault{\textminus}{TS1}
961 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
962 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
963 \DeclareTextSymbolDefault{\textasciigrave}{TS1}
964 \DeclareTextSymbolDefault{\texttildelow}{TS1}
965 \DeclareTextSymbolDefault{\textasciibreve}{TS1}
966 \DeclareTextSymbolDefault{\textasciicaron}{TS1}
967 \DeclareTextSymbolDefault{\textgravedbl}{TS1}
968 \DeclareTextSymbolDefault{\textacutedbl}{TS1}
969 \DeclareTextSymbolDefault{\textdagger}{TS1}
970 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
971 \DeclareTextSymbolDefault{\textbardbl}{TS1}
972 \DeclareTextSymbolDefault{\textperthousand}{TS1}
973 \DeclareTextSymbolDefault{\textbullet}{TS1}
974 \DeclareTextSymbolDefault{\textcelsius}{TS1}
975 \DeclareTextSymbolDefault{\textflorin}{TS1}
976 \DeclareTextSymbolDefault{\texttrademark}{TS1}
977 \DeclareTextSymbolDefault{\textcent}{TS1}
978 \DeclareTextSymbolDefault{\textsterling}{TS1}
979 \DeclareTextSymbolDefault{\textyen}{TS1}
```

```

980 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
981 \DeclareTextSymbolDefault{\textsection}{TS1}
982 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}
983 \DeclareTextSymbolDefault{\textcopyright}{TS1}
984 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
985 \DeclareTextSymbolDefault{\textlnot}{TS1}
986 \DeclareTextSymbolDefault{\textregistered}{TS1}
987 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
988 \DeclareTextSymbolDefault{\textdegree}{TS1}
989 \DeclareTextSymbolDefault{\textpm}{TS1}
990 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
991 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
992 \DeclareTextSymbolDefault{\textasciacute}{TS1}
993 \DeclareTextSymbolDefault{\textmu}{TS1}
994 \DeclareTextSymbolDefault{\textparagraph}{TS1}
995 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
996 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
997 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
998 \DeclareTextSymbolDefault{\textonequarter}{TS1}
999 \DeclareTextSymbolDefault{\textonehalf}{TS1}
1000 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
1001 \DeclareTextSymbolDefault{\texttimes}{TS1}
1002 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

1003 \DeclareTextCommandDefault{\texteuro}{%
1004 \CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```

1005 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}

```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```

1006 \DeclareTextCommandDefault{\textestimated}{%
1007 \tc@check@symbol3\textestimated}
1008 \DeclareTextCommandDefault{\textcurrency}{%
1009 \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.

```

1010 \DeclareTextCommandDefault{\capitaltie}{%
1011 \tc@check@accent2\capitaltie}
1012 \DeclareTextCommandDefault{\newtie}{%
1013 \tc@check@accent2\newtie}
1014 \DeclareTextCommandDefault{\capitalnewtie}{%
1015 \tc@check@accent2\capitalnewtie}
1016 \DeclareTextCommandDefault{\textleftarrow}{%
1017 \tc@check@symbol2\textleftarrow}
1018 \DeclareTextCommandDefault{\textrightarrow}{%
1019 \tc@check@symbol2\textrightarrow}
1020 \DeclareTextCommandDefault{\textblank}{%
1021 \tc@check@symbol2\textblank}
1022 \DeclareTextCommandDefault{\textdblhyphen}{%

```

```

1023 {\tc@check@symbol2{textdblhyphen}}
1024 \DeclareTextCommandDefault{\textzerooldstyle}{%
1025 {\tc@check@symbol2{textzerooldstyle}}
1026 \DeclareTextCommandDefault{\textoneoldstyle}{%
1027 {\tc@check@symbol2{textoneoldstyle}}
1028 \DeclareTextCommandDefault{\texttwooldstyle}{%
1029 {\tc@check@symbol2{texttwooldstyle}}
1030 \DeclareTextCommandDefault{\textthreeoldstyle}{%
1031 {\tc@check@symbol2{textthreeoldstyle}}
1032 \DeclareTextCommandDefault{\textfouroldstyle}{%
1033 {\tc@check@symbol2{textfouroldstyle}}
1034 \DeclareTextCommandDefault{\textfiveoldstyle}{%
1035 {\tc@check@symbol2{textfiveoldstyle}}
1036 \DeclareTextCommandDefault{\textsixoldstyle}{%
1037 {\tc@check@symbol2{textsixoldstyle}}
1038 \DeclareTextCommandDefault{\textsevenoldstyle}{%
1039 {\tc@check@symbol2{textsevenoldstyle}}
1040 \DeclareTextCommandDefault{\texteightoldstyle}{%
1041 {\tc@check@symbol2{texteightoldstyle}}
1042 \DeclareTextCommandDefault{\textnineoldstyle}{%
1043 {\tc@check@symbol2{textnineoldstyle}}
1044 \DeclareTextCommandDefault{\textangle}{%
1045 {\tc@check@symbol2{textangle}}
1046 \DeclareTextCommandDefault{\textrangle}{%
1047 {\tc@check@symbol2{textrangle}}
1048 \DeclareTextCommandDefault{\textmho}{%
1049 {\tc@check@symbol2{textmho}}
1050 \DeclareTextCommandDefault{\textbigcircle}{%
1051 {\tc@check@symbol2{textbigcircle}}
1052 \DeclareTextCommandDefault{\textuparrow}{%
1053 {\tc@check@symbol2{textuparrow}}
1054 \DeclareTextCommandDefault{\textdownarrow}{%
1055 {\tc@check@symbol2{textdownarrow}}
1056 \DeclareTextCommandDefault{\textborn}{%
1057 {\tc@check@symbol2{textborn}}
1058 \DeclareTextCommandDefault{\textdivorced}{%
1059 {\tc@check@symbol2{textdivorced}}
1060 \DeclareTextCommandDefault{\textdied}{%
1061 {\tc@check@symbol2{textdied}}
1062 \DeclareTextCommandDefault{\textleaf}{%
1063 {\tc@check@symbol2{textleaf}}
1064 \DeclareTextCommandDefault{\textmarried}{%
1065 {\tc@check@symbol2{textmarried}}
1066 \DeclareTextCommandDefault{\textmusicalnote}{%
1067 {\tc@check@symbol2{textmusicalnote}}
1068 \DeclareTextCommandDefault{\textdblhyphenchar}{%
1069 {\tc@check@symbol2{textdblhyphenchar}}
1070 \DeclareTextCommandDefault{\textdollaroldstyle}{%
1071 {\tc@check@symbol2{textdollaroldstyle}}
1072 \DeclareTextCommandDefault{\textcentoldstyle}{%
1073 {\tc@check@symbol2{textcentoldstyle}}
1074 \DeclareTextCommandDefault{\textcolonmonetary}{%
1075 {\tc@check@symbol2{textcolonmonetary}}
1076 \DeclareTextCommandDefault{\textwon}{%

```

```

1077 {\tc@check@symbol2{textwon}}
1078 \DeclareTextCommandDefault{\textnaira}{%
1079 {\tc@check@symbol2{textnaira}}
1080 \DeclareTextCommandDefault{\textguarani}{%
1081 {\tc@check@symbol2{textguarani}}
1082 \DeclareTextCommandDefault{\textpeso}{%
1083 {\tc@check@symbol2{textpeso}}
1084 \DeclareTextCommandDefault{\textlira}{%
1085 {\tc@check@symbol2{textlira}}
1086 \DeclareTextCommandDefault{\textrecipe}{%
1087 {\tc@check@symbol2{textrecipe}}
1088 \DeclareTextCommandDefault{\textinterrobang}{%
1089 {\tc@check@symbol2{textinterrobang}}
1090 \DeclareTextCommandDefault{\textinterrobangdown}{%
1091 {\tc@check@symbol2{textinterrobangdown}}
1092 \DeclareTextCommandDefault{\textdong}{%
1093 {\tc@check@symbol2{textdong}}
1094 \DeclareTextCommandDefault{\textpertenthousand}{%
1095 {\tc@check@symbol2{textpertenthousand}}
1096 \DeclareTextCommandDefault{\textpilcrow}{%
1097 {\tc@check@symbol2{textpilcrow}}
1098 \DeclareTextCommandDefault{\textbaht}{%
1099 {\tc@check@symbol2{textbaht}}
1100 \DeclareTextCommandDefault{\textnumero}{%
1101 {\tc@check@symbol2{textnumero}}
1102 \DeclareTextCommandDefault{\textdiscount}{%
1103 {\tc@check@symbol2{textdiscount}}
1104 \DeclareTextCommandDefault{\textopenbullet}{%
1105 {\tc@check@symbol2{textopenbullet}}
1106 \DeclareTextCommandDefault{\textservicemark}{%
1107 {\tc@check@symbol2{textservicemark}}
1108 \DeclareTextCommandDefault{\textlquill}{%
1109 {\tc@check@symbol2{textlquill}}
1110 \DeclareTextCommandDefault{\textrquill}{%
1111 {\tc@check@symbol2{textrquill}}
1112 \DeclareTextCommandDefault{\textcopyleft}{%
1113 {\tc@check@symbol2{textcopyleft}}
1114 \DeclareTextCommandDefault{\textcircledP}{%
1115 {\tc@check@symbol2{textcircledP}}
1116 \DeclareTextCommandDefault{\textreferencemark}{%
1117 {\tc@check@symbol2{textreferencemark}}
1118 \DeclareTextCommandDefault{\textsurd}{%
1119 {\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.

```

1120 \DeclareTextCommandDefault{\textcircled}{%
1121 {\CheckEncodingSubset\UseTextAccent{TS1}}%
1122 {\UseTextAccent{OMS}1\textcircled}
1123 \DeclareTextCommandDefault{\t}{%
1124 {\CheckEncodingSubset\UseTextAccent{TS1}}%
1125 {\UseTextAccent{OML}1\t}}

```

Finally input the encoding-specific definitions for TS1 thus making the top-level

definitions optimized for this encoding (and not for the default encoding).

```
1126 \input{ts1enc.def}
```

Now having the new glyphs available we also want to make sure that they are used. For most cases this will automatically happen but for some glyphs there are inferior definitions already known to L<sup>A</sup>T<sub>E</sub>X which will prevent the usage of the TS1 versions. So we better get rid of them:

```
1127 \UndeclareTextCommand{\textsterling}{OT1}
1128 \UndeclareTextCommand{\textdollar} {OT1}
```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```
1129 \%UndeclareTextCommand{\textsterling}{OT4}
1130 \%UndeclareTextCommand{\textdollar} {OT4}
```

From the T1 encoding there are two candidates for removal: %o and %oo since these are both constructed from % followed by a tiny ‘o’ rather than being a single glyph. The problem with this approach is that in PostScript fonts this small zero is usually not available resulting in %■ rather than %o 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 %o and %oo are not taken from the same physical font) and with PostScript fonts %oo will come out correctly while %oo 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.

```
1131 \UndeclareTextCommand{\textperthousand}{T1}
1132 \%UndeclareTextCommand{\textpertenthousand}{T1}
```

### 5.1.1 Supporting oldstyle digits

```
1133 \DeclareRobustCommand\oldstylenums[1]{%
1134 \begingroup
1135 \ifmmode
1136 \mathgroup\symletters #1%
1137 \else
1138 \CheckEncodingSubset\@use@text@encoding{TS1}%
1139 {\PackageWarning{textcomp}%
1140 {Oldstyle digits unavailable for
1141 family \f@family.\MessageBreak
1142 Lining digits used instead}}%
1143 \tw@{#1}%
1144 \fi
1145 \endgroup
1146 }
```

### 5.1.2 Subset encoding defaults

For many font families commonly used in the T<sub>E</sub>X 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:

```
1147 \iftc@forced \else
```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```
1148 \DeclareEncodingSubset{TS1}{cmr} {0}
1149 \DeclareEncodingSubset{TS1}{cmss} {0}
1150 \DeclareEncodingSubset{TS1}{cmtt} {0}
1151 \DeclareEncodingSubset{TS1}{cmvtt} {0}
1152 \DeclareEncodingSubset{TS1}{cmbr} {0}
1153 \DeclareEncodingSubset{TS1}{cmtl} {0}
1154 \DeclareEncodingSubset{TS1}{ccr} {0}
```

PSNFSS fonts:

```
1155 \DeclareEncodingSubset{TS1}{ptm} {4}
1156 \DeclareEncodingSubset{TS1}{pcr} {4}
1157 \DeclareEncodingSubset{TS1}{phv} {4}
1158 \DeclareEncodingSubset{TS1}{pp1} {3}
1159 \DeclareEncodingSubset{TS1}{pag} {4}
1160 \DeclareEncodingSubset{TS1}{pbk} {4}
1161 \DeclareEncodingSubset{TS1}{pnc} {4}
1162 \DeclareEncodingSubset{TS1}{pzc} {4}
1163 \DeclareEncodingSubset{TS1}{bch} {4}
1164 \DeclareEncodingSubset{TS1}{put} {5}
```

Other CTAN fonts (probably not complete):

```
1165 \DeclareEncodingSubset{TS1}{uag} {5}
1166 \DeclareEncodingSubset{TS1}{ugq} {5}
1167 \DeclareEncodingSubset{TS1}{u18} {4}
1168 \DeclareEncodingSubset{TS1}{u19} {4} % (LuxiSans, one day)
1169 \DeclareEncodingSubset{TS1}{augie} {5}
1170 \DeclareEncodingSubset{TS1}{dayrom} {3}
1171 \DeclareEncodingSubset{TS1}{dayroms} {3}
1172 \DeclareEncodingSubset{TS1}{pxr} {0}
1173 \DeclareEncodingSubset{TS1}{pxss} {0}
1174 \DeclareEncodingSubset{TS1}{pxtt} {0}
1175 \DeclareEncodingSubset{TS1}{txr} {0}
1176 \DeclareEncodingSubset{TS1}{txss} {0}
1177 \DeclareEncodingSubset{TS1}{txtt} {0}
```

Latin Modern and TeX Gyre:

```
1178 \DeclareEncodingSubset{TS1}{lmr} {0}
1179 \DeclareEncodingSubset{TS1}{lmdh} {0}
1180 \DeclareEncodingSubset{TS1}{lmss} {0}
1181 \DeclareEncodingSubset{TS1}{lmssq} {0}
1182 \DeclareEncodingSubset{TS1}{lmvtt} {0}
1183 \DeclareEncodingSubset{TS1}{lmtt} {0}
1184 \DeclareEncodingSubset{TS1}{qhv} {0}
1185 \DeclareEncodingSubset{TS1}{qag} {0}
1186 \DeclareEncodingSubset{TS1}{qbk} {0}
1187 \DeclareEncodingSubset{TS1}{qcr} {0}
1188 \DeclareEncodingSubset{TS1}{qcs} {0}
1189 \DeclareEncodingSubset{TS1}{qp1} {0}
1190 \DeclareEncodingSubset{TS1}{qtm} {0}
1191 \DeclareEncodingSubset{TS1}{qzc} {0}
1192 \DeclareEncodingSubset{TS1}{qhvc} {0}
```

Fourier-GUTenberg:

```
1193 \DeclareEncodingSubset{TS1}{futs} {4}
```

```

1194 \DeclareEncodingSubset{TS1}{futx} {4}
1195 \DeclareEncodingSubset{TS1}{futj} {4}
 Y&Y's Lucida Bright

```

```

1196 \DeclareEncodingSubset{TS1}{hlh} {3}
1197 \DeclareEncodingSubset{TS1}{hls} {3}
1198 \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`.

```

1199 \DeclareEncodingSubset{TS1}{hlct} {5}
1200 \DeclareEncodingSubset{TS1}{hlx} {5}
1201 \DeclareEncodingSubset{TS1}{hlce} {5}
1202 \DeclareEncodingSubset{TS1}{hlcn} {5}
1203 \DeclareEncodingSubset{TS1}{hlcw} {5}
1204 \DeclareEncodingSubset{TS1}{hlcf} {5}

```

Other commercial families...

```

1205 \DeclareEncodingSubset{TS1}{pplx} {3}
1206 \DeclareEncodingSubset{TS1}{pplj} {3}
1207 \DeclareEncodingSubset{TS1}{ptmx} {4}
1208 \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.

```

1209 \InputIfFileExists{textcomp.cfg}
1210 {\PackageInfo{textcomp}{Local configuration file used}{}{}}
1211 \fi
1212 {/TS1oldsty}

```

## 6 The `checkencodingsubset.tex` file

This is a simple file that asks for a name of a font family and then displays information about the TS1 encoding for this family and recommends the right encoding subset (to be used with `\DeclareEncodingSubset`) for this family.

```

1213 <*TS1check>
1214 \ProvidesFile{checkencodingsubset.tex}
1215 [2024/10/18 v0.5b Figure out safe TS1 encoding subsets]
1216 \let\typeoutdetails\typeout
1217 \%def\typeoutdetails#1{} % alternative definition used below

```

For the purpose of this check a glyph exists if the font slot is occupied—too bad if that contains the wrong glyph or some tofu. If it “exists” we return 0 otherwise 1. This way we can call this macro several times in a row and obtain a number that is 0 if all glyphs are existing or greater than 0 if any of them is missing.

The second argument (holding the command name for a symbol) is not used during these tests.

```

1218 \%def\doesglyphexist#1#2{\iffontchar\testFont #1 0\else 1\relax \fi}

```

This macro also tests and outputs some information about the symbol if it is missing. This time we make use of the second argument.

```

1219 \def\glyphmissingdetails#1#2{\iffontchar\testFont #1 \else
1220 \typeoutdetails{\space\space\space ==> \string#2 (#1) is missing}\fi}
1221 \newif\ifsafesubencodingfound
1222 \newif\ifcoremisses

```

Testing a group of symbols that belong to one sub-encoding. More precisely, the symbols that become unavailable if you change from sub-encoding  $x$  (#2) to  $x + 1$  (#3). As far as the code is concerned, the symbols that are supposed to be always available (the core) become available if we test the group -1 and 0.

The first argument contains the testing code and is supposed to return a single number greater or equal to zero.

```

1223 \def\testgroup#1#2#3{%
1224 \ifnum 0 = #1%
1225 \ifnum #2<0
1226 \typeoutdetails{All glyphs in core exist}%
1227 \else
1228 \typeoutdetails{All glyphs between sub-encoding #2 and #3 exist}%
1229 \fi
1230 \else
1231 \ifnum #2<0
1232 \typeoutdetails{*****}%
1233 \typeoutdetails{Some glyphs are missing from core:}%
1234 \coremissestrue
1235 \ifsafesubencodingfound \else
1236 \def\subencodingresult{#2}%
1237 \fi
1238 \else
1239 \typeoutdetails{Some glyphs are missing from sub-encoding #2:}%
1240 \ifsafesubencodingfound \else
1241 \def\subencodingresult{#3}%
1242 \fi
1243 \fi

```

If some glyphs are missing, we rerun the test code but this time using `\glyphmissingdetails`.

```
1244 {\let\doesglyphexist \glyphmissingdetails #1}%

```

And because we had misses we have definitely found the subset.

```

1245 \safesubencodingfoundtrue
1246 \fi
1247 }

```

The currently defined subset for the family is either stored in `\TS1:<family>` if it was declared, or it is the default subset which is stored in `\TS1:?`.

```
1248 \def\currsubencoding#1{\csname TS1:\ifcsname TS1:#1\endcsname #1\else ?\fi\endcsname}
```

If a font family is not found when declaring it with `\DeclareFixedFont` we end up with the following font. This can then be used as a simple test if we failed loading the `TS1` font.

```
1249 \DeclareFixedFont\cmrFont{TS1}{cmr}{m}{n}{10pt}
```

Check for all glyphs in all encoding subsets ...

```
1250 \def\testallgroups#1{%
1251 \DeclareFixedFont\testFont{TS1}{#1}{m}{n}{10pt}%
1252 \ifx\testFont\cmrFont
1253 \typeout{***** Font family #1 not found *****}%
1254 \else
```

We haven't checked anything yet.

```
1255 \safesubencodingfoundfalse
1256 \coremissesfalse
1257 \typeoutdetails{^^J-----}%
1258 \typeoutdetails{Testing font family #1^^J(currently TS1-sub-encoding
1259 \currsubencoding{#1})}%
1260 \typeout{-----}%
```

Then we start testing the groups beginning with the glyphs between sub-encoding 8 and 9. If any of them is missing (checked with \doesglyphexist) then we already know that 9 is the correct answer.

```
1261 \testgroup{%
1262 \doesglyphexist{21}{\texttwelveudash}%
1263 \doesglyphexist{22}{\textthreequartersemdash}%
1264 \doesglyphexist{134}{\textbardbl}%
1265 \doesglyphexist{137}{\textcelsius}%
1266 \doesglyphexist{178}{\texttwosuperior}%
1267 \doesglyphexist{179}{\textthreesuperior}%
1268 \doesglyphexist{185}{\textonesuperior}%
1269 }{8}{9}
```

Nevertheless we go on with further groups so that the output lists all missing glyphs.

```
1270 \testgroup{%
1271 \doesglyphexist{32}{\textblank}%
1272 \doesglyphexist{148}{\textinterrobang}%
1273 \doesglyphexist{149}{\textinterrobangdown}%
1274 \doesglyphexist{191}{\texteuro}%
1275 }{7}{8}
1276 \testgroup{%
1277 \doesglyphexist{47}{\textfractionsolidus}%
1278 \doesglyphexist{61}{\textminus}%
1279 \doesglyphexist{87}{\textohm}%
1280 \doesglyphexist{181}{\textmu}%
1281 }{6}{7}
1282 \testgroup{%
1283 \doesglyphexist{140}{\textflorin}%
1284 \doesglyphexist{164}{\textcurrency}%
1285 }{5}{6}
1286 \testgroup{%
1287 \doesglyphexist{155}{\textnumero}%
1288 \doesglyphexist{157}{\textestimated}%
1289 }{4}{5}
1290 \testgroup{%
1291 \doesglyphexist{24}{\textleftarrow}%
1292 \doesglyphexist{25}{\textrightarrow}%
1293 \doesglyphexist{94}{\textuparrow}%
1294 \doesglyphexist{95}{\textdownarrow}%
1295 \doesglyphexist{141}{\textcolonmonetary}%
}
```

```

1296 \doesglyphexist{142}{\textwon}%
1297 \doesglyphexist{146}{\textlira}%
1298 \doesglyphexist{150}{\textdong}%
1299 }{3}{4}%
1300 \testgroup{%
1301 \doesglyphexist{60}{\textangle}%
1302 \doesglyphexist{62}{\textrangle}%
1303 }{2}{3}%
1304 \testgroup{%
1305 \doesglyphexist{0}{\capitalgrave}%
1306 \doesglyphexist{1}{\capitalacute}%
1307 \doesglyphexist{2}{\capitalcircumflex}%
1308 \doesglyphexist{3}{\capitaltilde}%
1309 \doesglyphexist{4}{\capitaldieresis}%
1310 \doesglyphexist{5}{\capitalhungarumlaut}%
1311 \doesglyphexist{6}{\capitalring}%
1312 \doesglyphexist{7}{\capitalcaron}%
1313 \doesglyphexist{8}{\capitalbreve}%
1314 \doesglyphexist{9}{\capitalmacron}%
1315 \doesglyphexist{10}{\capitaldotaccent}%
1316 \doesglyphexist{11}{\capitalcedilla}%
1317 \doesglyphexist{12}{\capitalogonek}%
1318 \doesglyphexist{26}{\t}%
1319 \doesglyphexist{27}{\capitaltie}%
1320 \doesglyphexist{28}{\newtie}%
1321 \doesglyphexist{29}{\capitalnewtie}%
1322 \doesglyphexist{45}{\textdblhyphen}%
1323 \doesglyphexist{48}{\textzerooldstyle}%
1324 \doesglyphexist{49}{\textoneoldstyle}%
1325 \doesglyphexist{50}{\texttwooldstyle}%
1326 \doesglyphexist{51}{\textthreeoldstyle}%
1327 \doesglyphexist{52}{\textfouroldstyle}%
1328 \doesglyphexist{53}{\textfiveoldstyle}%
1329 \doesglyphexist{54}{\textsixoldstyle}%
1330 \doesglyphexist{55}{\textsevenoldstyle}%
1331 \doesglyphexist{56}{\texteightoldstyle}%
1332 \doesglyphexist{57}{\textnineoldstyle}%
1333 \doesglyphexist{77}{\textmho}%
1334 \doesglyphexist{79}{\textbigcircle}%
1335 \doesglyphexist{91}{\textlbrackdbl}%
1336 \doesglyphexist{93}{\textrbrackdbl}%
1337 \doesglyphexist{96}{\textasciigrave}%
1338 \doesglyphexist{98}{\textborn}%
1339 \doesglyphexist{99}{\textdivorced}%
1340 \doesglyphexist{100}{\textdied}%
1341 \doesglyphexist{108}{\textleaf}%
1342 \doesglyphexist{109}{\textmarried}%
1343 \doesglyphexist{110}{\textmusicalnote}%
1344 \doesglyphexist{126}{\texttildelow}%
1345 \doesglyphexist{127}{\textdblhyphenchar}%
1346 \doesglyphexist{128}{\textasciibreve}%
1347 \doesglyphexist{129}{\textasciicaron}%
1348 \doesglyphexist{175}{\textascimacron}%
1349 \doesglyphexist{130}{\textacutedbl}%

```

```

1350 \doesglyphexist{131}{\textgravedbl}%
1351 \doesglyphexist{138}{\textdollaroldstyle}%
1352 \doesglyphexist{139}{\textcentoldstyle}%
1353 \doesglyphexist{143}{\textnaira}%
1354 \doesglyphexist{144}{\textguarani}%
1355 \doesglyphexist{145}{\textpeso}%
1356 \doesglyphexist{147}{\textrecipe}%
1357 \doesglyphexist{152}{\textpertenthousand}%
1358 \doesglyphexist{153}{\textpilcrow}%
1359 \doesglyphexist{154}{\textbaht}%
1360 \doesglyphexist{156}{\textdiscount}%
1361 \doesglyphexist{158}{\textopenbullet}%
1362 \doesglyphexist{159}{\textservicemark}%
1363 \doesglyphexist{160}{\textlquill}%
1364 \doesglyphexist{161}{\textrquill}%
1365 \doesglyphexist{168}{\textasciidieresis}%
1366 \doesglyphexist{171}{\textcopyleft}%
1367 \doesglyphexist{173}{\textcircledP}%
1368 \doesglyphexist{180}{\textasciacute}%
1369 \doesglyphexist{184}{\textreferencemark}%
1370 \doesglyphexist{187}{\textsurd}%
1371 }{1}{2}%

```

All fonts (up to now) that belong to sub-encoding 1 do have the \textcircled glyph, but it is too small to be usable. So this test for this group currently doesn't do much good—but who knows maybe one day a font shows up in which this glyph is actually missing.

```

1372 \testgroup{%
1373 \doesglyphexist{79}{\textcircled}%
1374 this is not a proper test because the symbol is
1375 % usually available but not usable
1376 }{0}{1}%
1377 \testgroup{%
1378 \doesglyphexist{13}{\textquotestraightbase}%
1379 \doesglyphexist{18}{\textquotestraightdblbase}%
1380 \doesglyphexist{23}{\textcapitalcompwordmark}%
1381 \doesglyphexist{31}{\textascendercompwordmark}%
1382 \doesglyphexist{36}{\textdollar}%
1383 \doesglyphexist{39}{\textquotesingle}%
1384 \doesglyphexist{42}{\textasteriskcentered}%
1385 \doesglyphexist{132}{\textdagger}%
1386 \doesglyphexist{133}{\textdaggerdbl}%
1387 \doesglyphexist{135}{\textperthousand}%
1388 \doesglyphexist{136}{\textbullet}%
1389 \doesglyphexist{151}{\texttrademark}%
1390 \doesglyphexist{162}{\textcent}%
1391 \doesglyphexist{163}{\textsterling}%
1392 \doesglyphexist{165}{\textyen}%
1393 \doesglyphexist{166}{\textbrokenbar}%
1394 \doesglyphexist{167}{\textsection}%
1395 \doesglyphexist{169}{\textcopyright}%
1396 \doesglyphexist{170}{\textordfeminine}%
1397 \doesglyphexist{172}{\textlnot}%
1398 \doesglyphexist{174}{\textregistered}%
1399 \doesglyphexist{176}{\textdegree}%

```

```

1399 \doestglphexist{177}{\textpm}%
1400 \doestglphexist{182}{\textparagraph}%
1401 \doestglphexist{183}{\textperiodcentered}%
1402 \doestglphexist{186}{\textordmasculine}%
1403 \doestglphexist{188}{\textonequarter}%
1404 \doestglphexist{189}{\textonehalf}%
1405 \doestglphexist{190}{\textthreequarters}%
1406 \doestglphexist{214}{\texttimes}%
1407 \doestglphexist{246}{\textdiv}%
1408 }{-1}{0}%

```

If all groups have all glyphs then we have the full encoding (subset 0).

```

1409 \ifsafesubencodingfound\else
1410 \def\subencodingresult{0}%
1411 \fi

```

If the font is missing some of the core glyphs we make a remark about this, because they will never display.

```

1412 \typeoutdetails{-----}%
1413 \typeout{TS1 encoding subset for #1\ifcoremisses \space(ignoring core misses)\fi
1414 \space (\ifnum\subencodingresult =
1415 \currsubencoding{#1} ok\else bad\fi)}%
1416 \typeout{Use sub-encoding \subencodingresult
1417 \ifnum\subencodingresult = \currsubencoding{#1}\else
1418 \space (not \currsubencoding{#1})\fi}%
1419 \typeout{-----^~J}%
1420 \fi
1421 }

```

This tests all declarations (or most of them) that have been added to the kernel. It is called if no family is given interactively.

```

1422 \long\def\testallkerneldefinedfamilies{%
1423 \testallgroups{ccr}{0}
1424 \testallgroups{cmbr}{0}
1425 \% \testallgroups{cmr}{0} % don't test this one as it is the fallback
1426 % thus reports that the family is not found
1427 \testallgroups{cmss}{0}
1428 \testallgroups{cmtl}{0}
1429 \testallgroups{cmtt}{0}
1430 \testallgroups{cmvtt}{0}
1431 \testallgroups{pxr}{0}
1432 \testallgroups{pxss}{0}
1433 \testallgroups{pxtt}{0}
1434 \testallgroups{qag}{0}
1435 \testallgroups{qbk}{0}
1436 \testallgroups{qcr}{0}
1437 \testallgroups{qcs}{0}
1438 \testallgroups{qhvc}{0}
1439 \testallgroups{qhvh}{0}
1440 \testallgroups{qpl}{0}
1441 \testallgroups{qtm}{0}
1442 \testallgroups{qzc}{0}
1443 \testallgroups{txr}{0}
1444 \testallgroups{txss}{0}
1445 \testallgroups{txtt}{0}

```

```

1446 %
1447 % Next would claim to be 0 (or 2)
1448 %
1449 \%testallgroups{lmr}%
1450 \%testallgroups{lmdh}%
1451 \%testallgroups{lmss}%
1452 \%testallgroups{lmssq}%
1453 \%testallgroups{lmvtt}%
1454 \%testallgroups{lmtt}%
1455 {1} % missing TM, SM and pertenthousand so really 2
1456 %
1457 % these are no longer in TeX Live
1458 %
1459 \%testallgroups{ptmx}%
1460 \%testallgroups{ptmj}%
1461 \%testallgroups{ul8}%
1462 %
1463 % next block has tofu chars so results are wrong
1464 \%testallgroups{bch}%
1465 \%testallgroups{futj}%
1466 \%testallgroups{futs}%
1467 \%testallgroups{futx}%
1468 \%testallgroups{pag}%
1469 \%testallgroups{pbk}%
1470 \%testallgroups{pcr}%
1471 \%testallgroups{phv}%
1472 \%testallgroups{pnc}%
1473 \%testallgroups{pplj}%
1474 \%testallgroups{pplx}%
1475 \%testallgroups{pppl}%
1476 \%testallgroups{ptm}%
1477 \%testallgroups{pzcl}%
1478 \%testallgroups{ul19}%
1479 \%testallgroups{dayroms}%
1480 \%testallgroups{dayrom}%
1481 \%testallgroups{augie}%
1482 \%testallgroups{put}%
1483 \%testallgroups{uag}%
1484 \%testallgroups{ugq}%
1485 %
1486 \testallgroups{zi4}%
1487 %
1488 %% not installed normally
1489 %
1490 \%testallgroups{hls}%
1491 \%testallgroups{hlst}%
1492 \%testallgroups{hlct}%
1493 \%testallgroups{hlh}%
1494 \%testallgroups{hlx}%
1495 \%testallgroups{hlce}%
1496 \%testallgroups{hlcn}%
1497 \%testallgroups{hlcw}%
1498 \%testallgroups{hlcif}%
1499

```

```

1500 \testallgroups{lato-LF}{} {0} % with a bunch of tofu inside --- should probably be changed
1501 \testallgroups{opensans-TLF}{}{0} % with a bunch of tofu inside --- should probably be change
1502 \testallgroups{cantarell-TLF}{} {0} % with a bunch of tofu inside --- should probably be cha
1503 \testallgroups{fbb-LF}{} {0} % missing centoldstyle ---> 2
1504 \testallgroups{tli}{} {1} % with lots of tofu inside --- should probably be changed
1505 \testallgroups{Alegreya-OsF}{} {2}
1506 \testallgroups{AlegreyaSans-OsF}{} {2}
1507 \testallgroups{DejaVuSans-TLF}{} {2}
1508 \testallgroups{DejaVuSansCondensed-TLF}{} {2}
1509 \testallgroups{DejaVuSansMono-TLF}{} {2} this is missing \textfractionsolidus which makes it 7
1510 \testallgroups{EBGaramond-LF}{} {2}
1511 \testallgroups{Tempora-TLF}{} {2}
1512 \testallgroups{Tempora-T0sF}{} {2}
1513 \testallgroups{Arimo-TLF}{} {3}
1514 \testallgroups{Crlt-TLF}{} {3} changed from Carlito-
1515 \testallgroups{FiraSans-LF}{} {3} should be 4
1516 \testallgroups{IBMPlexSans-TLF}{} {3}
1517 \testallgroups{Merriwthr-OsF}{} {3} changed from Merriweather- and should be 2
1518 \testallgroups{Montserrat-LF}{} {3} now 2
1519 \testallgroups{MontserratAlternates-LF}{}{3} now 2
1520 \testallgroups{SourceCodePro-TLF}{} {3}
1521 \testallgroups{SourceCodePro-T0sF}{} {3}
1522 \testallgroups{SourceSansPro-OsF}{} {3}
1523 \testallgroups{SourceSerifPro-LF}{} {3}
1524 \testallgroups{Tinos-TLF}{} {3}
1525 \testallgroups{AccanthisADFSStdNoThree-LF}{}{4}
1526 \testallgroups{Cabin-TLF}{} {4}
1527 \testallgroups{Caladea-TLF}{} {4}
1528 \testallgroups{Chivo-LF}{} {4}
1529 \testallgroups{ClearSans-TLF}{} {4}
1530 \testallgroups{Coelacanth-LF}{} {4}
1531 \testallgroups{CrimsonPro-LF}{} {4}
1532 \testallgroups{FiraMono-TLF}{} {4}
1533 \testallgroups{FiraMono-T0sF}{} {4}
1534 \testallgroups{Go-TLF}{} {4}
1535 \testallgroups{GoMono-TLF}{} {4}
1536 \testallgroups{InriaSans-LF}{} {4}
1537 \testallgroups{InriaSerif-LF}{} {4}
1538 \testallgroups{LibertinusSans-LF}{} {4}
1539 \testallgroups{LibertinusSerif-LF}{} {4}
1540 \testallgroups{LibreBodoni-TLF}{} {4}
1541 \testallgroups{LibreFranklin-TLF}{} {4}
1542 \testallgroups{LinguisticsPro-LF}{} {4}
1543 \testallgroups{LinguisticsPro-OsF}{} {4}
1544 \testallgroups{LinuxBiolinumT-LF}{} {4}
1545 \testallgroups{LinuxLibertineT-LF}{} {4}
1546 \testallgroups{MerriwthrSans-OsF}{} {4} name change and now 2
1547 \testallgroups{MintSpirit-LF}{} {4}
1548 \testallgroups{MintSpiritNoTwo-LF}{} {4}
1549 \testallgroups{PTMono-TLF}{} {4}
1550 \testallgroups{PTSans-TLF}{} {4}
1551 \testallgroups{PTSansCaption-TLF}{} {4}
1552 \testallgroups{PTSansNarrow-TLF}{} {4}
1553 \testallgroups{PTSerif-TLF}{} {4}

```

```

1554 \testallgroups{PTSerifCaption-TLF}%
1555 \testallgroups{Raleway-TLF}%
1556 \testallgroups{Raleway-T0sF}%
1557 \testallgroups{Roboto-LF}%
1558 \testallgroups{RobotoMono-TLF}%
1559 \testallgroups{RobotoSlab-TLF}%
1560 \testallgroups{Rosario-LF}%
1561 \testallgroups{SticksTooText-LF}%
1562 \testallgroups{UniversalisADFStd-LF}%
1563 \testallgroups{Almnndr-OsF}%
1564 \testallgroups{Baskervaldux-LF}%
1565 \testallgroups{BaskervilleF-LF}%
1566 \testallgroups{Bttr-TLF}%
1567 \testallgroups{Cinzel-LF}%
1568 \testallgroups{CinzelDecorative-LF}%
1569 \testallgroups{DejaVuSerif-TLF}%
1570 \testallgroups{DejaVuSerifCondensed-TLF}%
1571 \testallgroups{GilliusADF-LF}%
1572 \testallgroups{charssil-TLF}%
1573 \testallgroups{GilliusADFCond-LF}%
1574 \testallgroups{GilliusADFNoTwo-LF}%
1575 \testallgroups{GilliusADFNoTwoCond-LF}%
1576 \testallgroups{Lbstr-LF}%
1577 \testallgroups{OldStandard-TLF}%
1578 \testallgroups{PlyfrDisplay-LF}%
1579 \testallgroups{PlyfrDisplay-OsF}%
1580 \testallgroups{TheanoDidot-TLF}%
1581 \testallgroups{TheanoDidot-T0sF}%
1582 \testallgroups{TheanoModern-TLF}%
1583 \testallgroups{TheanoModern-T0sF}%
1584 \testallgroups{TheanoOldStyle-TLF}%
1585 \testallgroups{TheanoOldStyle-T0sF}%
1586 \testallgroups{Crimson-TLF}%
1587 \testallgroups{IBMPlexMono-TLF}%
1588 \testallgroups{IBMPlexSerif-TLF}%
1589 \testallgroups{LibertinusMono-TLF}%
1590 \testallgroups{LibertinusSerifDisplay-LF}%
1591 \testallgroups{LinuxLibertineDisplayT-LF}%
1592 \testallgroups{LinuxLibertineMonoT-LF}%
1593 \testallgroups{LinuxLibertineMonoT-TLF}%
1594 \testallgroups{Ovrlck-LF}%
1595 \testallgroups{CormorantGaramond-LF}%
1596 \testallgroups{Heuristica-TLF}%
1597 \testallgroups{Heuristica-T0sF}%
1598 \testallgroups{IMFELLEnglish-TLF}%
1599 \testallgroups{LibreBskvl-LF}%
1600 \testallgroups{LibreCsln-LF}%
1601 \testallgroups{Mrcls-LF}%
1602 \testallgroups{NotoSans-LF}%
1603 \testallgroups{NotoSansMono-TLF}%
1604 \testallgroups{NotoSansMono-T0sF}%
1605 \testallgroups{NotoSerif-LF}%
1606 \testallgroups{Quattro-LF}%
1607 \testallgroups{QuattroSans-LF}%

```

```

1608 \testallgroups{XCharter-TLF}%
1609 \testallgroups{XCharter-T0sF}%
1610 \testallgroups{erewhon-LF}%
1611 \testallgroups{ComicNeue-TLF}%
1612 \testallgroups{ComicNeueAngular-TLF}%
1613 \testallgroups{Frm-LF}%
1614 \testallgroups{Cochineal-TLF}%
1615 \testallgroups{AlgolRevived-TLF}%
1616 }

There interaction with the user.

1617 \typeout{^^J=====
1618 \typeout{| Enter font family to check (or <enter> for kernel defined families)}
1619 \typeout{=====}
1620 \typein[\FontFamilyToCheck]{}

1621 \if!\FontFamilyToCheck!
1622 \typeout{=====}
1623 \typeout{| Detailed output? (default no)}
1624 \typeout{=====}
1625 \typein[\Details]{}
1626 \if!\Details!
1627 \def\typeoutdetails#1{}
1628 \else
1629 \let\typeoutdetails\typeout
1630 \fi
1631 \testallkerneldefinedfamilies
1632 \else
1633 \let\typeoutdetails\typeout
1634 \testallgroups\FontFamilyToCheck
1635 \fi

1636 \stop
1637
```

## File 34

# 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 page number style with the `\pagenumbering{<foo>}` command, which sets the page counter to 1 and defines `\thepage` to be `\foo`. For example, `\pagenumbering{roman}` causes pages to be numbered i, ii, etc.

```
1 {*2ekernel}
2 \message{page nos.,}
3 \countdef\c@page=0 \c@page=1
4 \def\cl@page{}
5 \def\pagenumbering#1{%
6 \global\c@page \cne \gdef\thepage{\csname \#1\endcsname
7 \c@page}}
8 {/2ekernel}
```

# File 35

## ltxref.dtx

### 1 Cross Referencing

The user writes `\label{<foo>}` to define the following cross-references:

`\ref*<{<foo>}`: value of most recently incremented referenceable counter. in the current environment. (Chapter, section, theorem, footnote and enumeration counters and other counters stepped with `\refstepcounter` are referenceable.)

`\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.

Note: the starred versions `\ref*` and `\pageref*` are provided to align with the use of `hyperref`. Without `hyperref` (or some other package using the starred form) the star is simply ignored.

Note: starting with 2023-06-01 `\label` stores also the current value of `\@currentlabelname` which should typically contain a (sanitized) title. (A reference command `\nameref` is provided by the `nameref` package.) `\label` also stores `\@currentHref` which if set should refer to a target name for links. This value is set and used by `hyperref`. Unlike the other values `\@currentHref` should be set globally. A fifth value `\@kernel@reserved@label@data` is reserved for the kernel to allow future extensions of the cross-reference system.

#### 1.1 Cross Referencing

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
1 <*2ekernel>
2 \message{x-ref,}
```

This is implemented as follows. A referenceable 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)}%
```

```
{eval(\@currentlabelname)}{eval(\@currentHref)}{eval(\@kernel@reserved@label@data)}}

\ref{FOO} ==
BEGIN
 if \r@foo undefined
 then @refundefined := G T
 ??
 Warning: 'reference foo on page ... undefined'
```

```

 else \@car \eval(\r@FOO)\@nil
 fi
END

\pageref{foo} =
BEGIN
if \r@foo undefined
then @refundefined := G T
??
Warning: 'reference foo on page ... undefined'
else \@cdr \eval(\r@FOO)\@nil
fi
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 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}{\'abra-\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.

```

3 % \newif\ifG@refundefined
4 % \def\G@refundefinedtrue{\global\let\ifG@refundefined\iftrue}
5 % \def\G@refundefinedfalse{\global\let\ifG@refundefined\iffalse}
6 \def\G@refundefinedtrue{%
7 \gdef\@refundefined{%
8 @latex@warning@no@line{There were undefined references}}}
9 \let\@refundefined\relax

```

*(End of definition for `\G@refundefinedtrue` and `\@refundefined`.)*

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\ref</code>     | Referencing a <code>\label</code> . RmS 91/10/25: added a few extra <code>\reset@font</code> , as suggested by Bernd Raichle                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>\pageref</code> | RmS 92/08/14: made <code>\ref</code> and <code>\pageref</code> robust<br>RmS 93/09/08: Added setting of <code>refundefined</code> switch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>\@setref</code> | <pre> 10 &lt;/2ekernel&gt; 11 &lt;*2ekernel   latexrelease&gt; 12 &lt;latexrelease&gt;\IncludeInRelease{2023/06/01}% 13 &lt;latexrelease&gt;          {\@kernel@sref}{store five arguments}% 14 \def\@setref#1#2#3{% 15   \ifx#1\relax 16     \protect\G@refundefinedtrue 17     \nfss@text{\reset@font\bfseries ??}% 18     \@latex@warning{Reference '#3' on page \thepage \space 19                   undefined}% 20   \else 21     \expandafter#2#1\empty\empty\empty\null 22   \fi} 23 \long\def\@firstoffive#1#2#3#4#5{#1} 24 \long\def\@secondoffive#1#2#3#4#5{#2} 25 \def\@kernel@sref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoffive{#1}} 26 \def\@kernel@spageref#1{\expandafter\@setref\csname r@#1\endcsname 27           \@secondoffive{#1}} 28 &lt;latexrelease&gt;\EndIncludeInRelease 29 &lt;latexrelease&gt;\IncludeInRelease{2022/06/01}% 30 &lt;latexrelease&gt;          {\@kernel@sref}{store five arguments}% 31 &lt;latexrelease&gt;\def\@setref#1#2#3{% 32   &lt;latexrelease&gt; \ifx#1\relax 33     &lt;latexrelease&gt; \protect\G@refundefinedtrue 34     &lt;latexrelease&gt; \nfss@text{\reset@font\bfseries ??}% 35     &lt;latexrelease&gt; \@latex@warning{Reference '#3' on page \thepage \space 36                   undefined}% 37   &lt;latexrelease&gt; \else 38   &lt;latexrelease&gt; \expandafter#2#1\null 39   &lt;latexrelease&gt; \fi} 40 &lt;latexrelease&gt;\let\@firstoffive\undefined 41 &lt;latexrelease&gt;\let\@secondoffive\undefined 42 &lt;latexrelease&gt;\def\@kernel@sref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}} 43 &lt;latexrelease&gt;\def\@kernel@spageref#1{\expandafter\@setref\csname r@#1\endcsname 44           \@secondoftwo{#1}} 45 &lt;latexrelease&gt;\EndIncludeInRelease 46 &lt;latexrelease&gt;\IncludeInRelease{0000/00/00}% </pre> |

```

47 \begin{macro}{\@kernel@sref}{store five arguments}%
48 \def\@setref#1#2#3{%
49 \ifx#1\relax
50 \protect\G@refundefinedtrue
51 \nfss@text{\reset@font\bfseries ??}%
52 \@latex@warning[Reference '#3' on page \thepage \space
53 undefined]%
54 \else
55 \expandafter#2#1\null
56 \fi}
57 \let\@firstoffive\undefined
58 \let\@secondoffive\undefined
59 \let\@kernel@sref\undefined
60 \let\@kernel@spageref\undefined
61 \EndIncludeInRelease
62 \IncludeInRelease{2022/06/01}%
63 \begin{macro}{\@ref}{Add starred reference commands}%
64 \let\@kernel@ref\@kernel@sref
65 \let\@kernel@pageref\@kernel@spageref
66 \NewDocumentCommand{\ref}{s}
67 {\IfBooleanTF{#1}{\@kernel@sref}{\@kernel@ref}}
68 \NewDocumentCommand{\pageref}{s}
69 {\IfBooleanTF{#1}{\@kernel@spageref}{\@kernel@pageref}}%

```

As the commands are now protected we also need expandable versions for use in `\ifthenelse`:

```

70 \def\@kernel@pageref@exp#1{\csname cs_if_exist:cTF\endcsname
71 {r@#1}\{\csname tl_item:cn\endcsname{r@#1}{2}\}{0}\}
72 \def\@kernel@ref@exp#1{\csname cs_if_exist:cTF\endcsname
73 {r@#1}\{\csname tl_item:cn\endcsname{r@#1}{1}\}{0}\}
74 (/2ekernel | \begin{macro}{\@kernel}{\@release})
75 \EndIncludeInRelease
76 \IncludeInRelease{0000/00/00}%
77 \begin{macro}{\@ref}{Add starred reference commands}%
78 \def\@ref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}}
79 \def\@pageref#1{\expandafter\@setref\csname r@#1\endcsname
80 \@secondoftwo{#1}}
81 \begin{macro}{\@pageref}{\@ref}
82 \EndIncludeInRelease
83 (*2ekernel)

```

(End of definition for `\ref`, `\pageref`, and `\@setref`.)

**\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 `\bibcrite`. 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.

```

84 \def\@newl@bel#1#2#3{%
85 \@ifundefined{#1#2}%
86 \relax
87 \gdef\@multiplelabels{%
88 \@latex@warning@no@line{There were multiply-defined labels}}%

```

```

89 \@latex@warning@no@line{Label '#2' multiply defined}%
90 \global\@namedef{\@newl@bel r}{}
91 \def\newlabel{\@newl@bel r}
92 \onlypreamble\@newl@bel

```

(End of definition for `\newlabel` and `\@newl@bel`.)

`\if@multiplelabels` This is redefined to produce a warning if at least one label is defined more than once. It is executed by the `\enddocument` command.

```

93 \let \@multiplelabels \relax

```

(End of definition for `\if@multiplelabels` and `\@multiplelabels`.)

`\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.

```

94 </2ekernel>
95 <*2ekernel | latexrelease>
96 <latexrelease>\IncludeInRelease{2023/06/01}%
97 <latexrelease> {\label}{store five label arguments}%
98 \providecommand{\currentlabelname}{}
99 \providecommand{\currentHref}{}
100 \providecommand{\kernel@reserved@label@data}{}
101 \NewHookWithArguments{label}{1}
102 \def\label#1{\@bsphack
103 \begingroup
104 \UseHookWithArguments{label}{1}{#1}%
105 \protected@write{\auxout}{}
106 {\string\newlabel{#1}{\currentlabelname{\thepage}}%
107 {\currentlabelname{\currentHref}{\kernel@reserved@label@data}}}%
108 \endgroup
109 \@esphack
110 <|latexrelease>\EndIncludeInRelease

```

(End of definition for `\label`. This function is documented on page 750.)

`\refstepcounter` Step the counter and allow for labels to point to its current value.

```

111 <|latexrelease>\IncludeInRelease{2022/06/01}%
112 <|latexrelease> {\Ref}{Add starred version}%
113 \def\@currentcounter{}%
114 <|latexrelease>\EndIncludeInRelease
115 <|latexrelease>\IncludeInRelease{2024/11/01}%
116 <|latexrelease> {\@currentHref}{set theHcounter representation}%

```

`refstepcounter (socket)` This socket takes the whole code as argument. The default kernel plug is identity. By changing the plug hyperref can add a conditional and e.g. suppress the processing in a PDF context.

```

117 \NewSocket{refstepcounter}{1}

```

`refstepcounter/target (socket)` This socket takes an argument, the counter name, and should at least set from it the target name `\@currentHref`. With `hyperref` it sets also the actual target. This is done with a socket so that the target name is not set more than once to (possibly) different names. The socket is not used in `\@kernel@refstepcounter`. The tagging code needs the target name so it is added after this socket.

```
118 \NewSocket{refstepcounter/target}{1}
```

`(refstepcounter/target) (plug)`

```
119 \NewSocketPlug{refstepcounter/target}{kernel}
120 {\xdef\@currentHref {\#1.\csname theH\#1\endcsname}}%
121 \AssignSocketPlug{refstepcounter/target}{kernel}

122 \def\refstepcounter#1{%
123 \UseSocket{refstepcounter}{%
124 \stepcounter{#1}%
125 \edef\@currentcounter{#1}%
126 \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.

```
127 {\csname p@\#1\expandafter\endcsname\csname the\#1\endcsname}%
128 \UseSocket{refstepcounter/target}{#1}%
129 \UseTaggingSocket{recordtarget}%
130 }%
131 }
```

This is a version of `\refstepcounter` which does not set and use targets.

```
\@kernel@refstepcounter
132 \def\@kernel@refstepcounter#1{%
133 \UseSocket{refstepcounter}{%
134 \stepcounter{#1}%
135 \edef\@currentcounter{#1}%
136 \protected@edef\@currentlabel
137 {\csname p@\#1\expandafter\endcsname\csname the\#1\endcsname}}}%
138 <|latexrelease>\EndIncludeInRelease
139 <|latexrelease>\IncludeInRelease{2022/06/01}%
140 <|latexrelease> {\@currentHref}{set theHcounter representation}%
141 <|latexrelease>\def\refstepcounter#1{\stepcounter{#1}%
142 <|latexrelease> \edef\@currentcounter{#1}%
143 <|latexrelease> \protected@edef\@currentlabel
144 <|latexrelease> {\csname p@\#1\expandafter\endcsname\csname the\#1\endcsname}}%
145 <|latexrelease>%
146 <|latexrelease>\let\@kernel@refstepcounter\refstepcounter
147 <|latexrelease>\EndIncludeInRelease
```

(End of definition for \refstepcounter and \kernel@refstepcounter.)

```
148 <latexrelease>\IncludeInRelease{2022/06/01}%
149 <latexrelease> {\Ref}{Add starred version}%
```

\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.

```
150 \def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

(End of definition for \labelformat.)

\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.<sup>39</sup>

```
151 \def\kernel@Ref#1{\protected@edef\tempa{\kernel@ref{#1}}%
152 \expandafter\MakeUppercase\tempa}
153 \def\kernel@sRef#1{\protected@edef\tempa{\kernel@sref{#1}}%
154 \expandafter\MakeUppercase\tempa}
155 \NewDocumentCommand\Ref{s}
156 {\IfBooleanTF{#1}{\kernel@sRef}{\kernel@Ref}}
```

(End of definition for \Ref.)

```
157 </2ekernel | latexrelease>
158 <latexrelease>\EndIncludeInRelease
159 <latexrelease>\IncludeInRelease{0000/00/00}%
160 <latexrelease> {\label}{store five label arguments}%
161 <latexrelease>\let\currenttitle\undefined
162 <latexrelease>\let\currenttarget\undefined
163 <latexrelease>\let\currentdata\undefined
164 <latexrelease>\def\label#1{\@bsphack
165 \protected@write\auxout{}%
166 {\string\newlabel{#1}{{\currentlabel}{\thepage}}}%
167 \@esphack}
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{2020/10/01}%
170 <latexrelease> {\Ref}{Add starred version}%
171 <latexrelease>\def\currentcounter{}
172 <latexrelease>\def\refstepcounter#1{\stepcounter{#1}%
173 \edef\currentcounter{#1}%
174 \protected@edef\currentlabel
175 {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
176 }
177 <latexrelease>\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
178 <latexrelease>\DeclareRobustCommand\Ref[1]{\protected@edef\tempa{\ref{#1}}%
179 \expandafter\MakeUppercase\tempa}
180 <latexrelease>\EndIncludeInRelease
181 <latexrelease>\IncludeInRelease{2019/10/01}%
182 <latexrelease> {\refstepcounter}{Add \labelformat and \Ref}%
183 <latexrelease>\let\currentcounter\undefined
```

---

<sup>39</sup>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.

```

184 〈latexrelease〉\def\refstepcounter#1{\stepcounter{#1}%
185 〈latexrelease〉 \protected@edef\@currentlabel
186 〈latexrelease〉 {\csname p@\#1\expandafter\endcsname\csname the\#1\endcsname}%
187 〈latexrelease〉}
188 〈latexrelease〉\def\labelformat#1{\expandafter\def\csname p@\#1\endcsname##1}%
189 〈latexrelease〉\DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%
190 〈latexrelease〉 \expandafter\MakeUppercase\@tempa}
191 〈latexrelease〉\EndIncludeInRelease
192 〈latexrelease〉\IncludeInRelease{0000/00/00}%
193 〈latexrelease〉 {\refstepcounter}{Add \labelformat and \Ref}%
194 〈latexrelease〉
195 〈latexrelease〉\def\refstepcounter#1{\stepcounter{#1}%
196 〈latexrelease〉 \protected@edef\@currentlabel
197 〈latexrelease〉 {\csname p@\#1\endcsname\csname the\#1\endcsname}%
198 〈latexrelease〉}
199 〈latexrelease〉\let\labelformat\@undefined
200 〈latexrelease〉\let\Ref\@undefined
201 〈latexrelease〉
202 〈latexrelease〉\EndIncludeInRelease
203 〈*2ekernel〉

```

**\@currentlabel** Default for \label commands that come before any environment.

```
204 \def\@currentlabel{}%
```

(*End of definition for \@currentlabel.*)

```
205 〈/2ekernel〉
```

# File 36

## ltproperties.dtx

### Abstract

This code implements command to record and (expandably) reference document properties. It extends the standard `\label/\ref/\pageref` commands.

## 1 Introduction

The module allows to record the “current state” of various document properties (typically the content of macros and values of counters) and to access them in other places through a label. The list of properties that can be recorded and retrieved are not fix and can be extended by the user. The values of the properties are recorded in the `.aux` file and can be retrieved at the second compilation.

The module uses the ideas of properties and labels. A label is a document reference point: a name for the user. An property is something that L<sup>A</sup>T<sub>E</sub>X can track, such as a page number, section number or name. The names of labels and properties may be arbitrary. Note that there is a single namespace for each.

## 2 Design discussion

The design here largely follows ideas from `zref`. In particular, there are two independent concepts: properties that can be recorded between runs, and labels which consist of lists of these properties. The reason for the split is that individual labels will want to record some but not all properties. For examples, a label concerned with position would track the *x* and *y* coordinates of the current point, but not for example the page number.

In the current implementation, properties share a single namespace. This allows multiple lists to re-use the same properties, for example page number, absolute page number, etc. This does mean that *changing* a standard property is an issue. However, some properties have complex definitions (again, see `zref` at present): having them in a single shared space avoids the need to copy code.

Labels could be implemented as prop data. That is not done at present as there is no obvious need to map to or copy the data. As such, faster performance is available using a hash table approach as in a “classical” set up. Data written to the `.aux` file uses simple paired *balanced text* not keyvals: this avoids any restrictions on names and again offers increased performance.

The `expl3` versions of the label command do not use `\@bsphack/\@esphack` to avoid double spaces, but the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> command does as it lives at the document command level.

The reference commands are expandable.

Currently the code has nearly no impact on the main `\label` and `\ref` commands as too many external packages rely on the concrete implementation. There is one exception: the label names share the same namespace. That means that if both `\label{ABC}` and `\RecordProperties{ABC}{page}` are used there is a warning `Label ‘ABC’ multiply defined`.

### 3 Handling unknown labels and properties

With the standard `\label/\ref` commands the requested label is either in the `.aux`-file (and so known) or not. In the first case the stored value can be used, in the second case the reference commands print two question marks.

With flexible property lists a reference commands asks for the value of a specific property stored under a label name and we have to consider more variants:

- If the requested property is unknown (not declared) the system is not correctly set up and an error is issued.
- If the label is unknown, the default of the property is used.
- If the label is known, but doesn't provide a value for the property then again the default of the property is used.
- The command `\property_ref:nnn` allows to give a local default which is used instead of the property default in the two cases before.

### 4 Rerun messages

As the reference commands are expandable they can neither issue a message that the label or the label-property combination is unknown, nor can they trigger the rerun message at the end of the L<sup>A</sup>T<sub>E</sub>X run.

Where needed such messages must therefore be triggered manually. For this two commands are provided: `\property_ref_undefined_warn:` and `\property_ref_-undefined_warn:nn`. See below for a description.

### 5 Open points

- The `xpos` and `ypos` properties require that the position is stored first but there is no (public) engine independent interface yet. Code must use `\tex_savepos:D`.

### 6 Code interfaces

---

```
\property_new:nnnn \property_new:nnnn {<property>} {<setpoint>} {<default>} {<code>}
\property_gset:nnnn \property_gset:nnnn {<property>} {<setpoint>} {<default>} {<code>}
```

L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub>-interface: see `\NewProperty`, `\SetProperty`.

Sets the `<property>` to have the `<default>` specified, and at the `<setpoint>` (either now or `shipout`) to write the result of the `<code>` as part of a label. The `<code>` should be expandable. The expansion of `<code>` (the value of the property) is written to the `.aux` file and read back from there at the next compilation. Values should assume that the standard L<sup>A</sup>T<sub>E</sub>X catcode régime with @ a letter is active then.

If the property is declared within a package it is suggested that its name is build from letters, hyphens and slashes, and is always structured as follows:

`<package-name>/<property-name>`.

|                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\property_record:nN</code>             | <code>\property_record:nN {&lt;label&gt;} {clist var}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>\property_record:nn</code>             | <code>\property_record:nn {&lt;label&gt;} {&lt;clist&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>\property_record:(nV ee)</code>        | <p><b>LATEX 2<sub>E</sub>-interface:</b> see <code>\RecordProperties</code>.</p> <p>Writes the list of properties given by the <code>&lt;clist&gt;</code> to the <code>.aux</code> file with the <code>&lt;label&gt;</code> specified.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>\property_ref:nn *</code>              | <code>\property_ref:nn {&lt;label&gt;} {&lt;property&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>\property_ref:ee *</code>              | <p><b>LATEX 2<sub>E</sub>-interface:</b> see <code>\RefProperty</code>.</p> <p>Expands to the value of the <code>&lt;property&gt;</code> for the <code>&lt;label&gt;</code>, if available, and the default value of the property otherwise. If <code>&lt;property&gt;</code> has not been declared with <code>\property_new:nnnn</code> an error is issued. The command raises an internal, expandable, local flag if the reference can not be resolved.</p>                                                                                                                                                                                                                         |
| <code>\property_ref:nnn *</code>             | <code>\property_ref:nnn {&lt;label&gt;} {&lt;property&gt;} {&lt;local default&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>\property_ref:een *</code>             | <p><b>LATEX 2<sub>E</sub>-interface:</b> see <code>\RefProperty</code>.</p> <p>Expands to the value of the <code>&lt;property&gt;</code> for the <code>&lt;label&gt;</code>, if available, and to <code>&lt;local default&gt;</code> otherwise. If <code>&lt;property&gt;</code> has not been declared with <code>\property_new:nnnn</code> an error is issued. The command raises an internal, expandable local flag if the reference can not be resolved.</p>                                                                                                                                                                                                                      |
| <code>\property_ref undefined_warn:</code>   | <code>\property_ref undefined_warn:</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|                                              | <p><b>LATEX 2<sub>E</sub>-interface:</b> not provided.</p> <p>The commands triggers the standard warning</p> <p><b>LaTeX Warning:</b> <code>There were undefined references.</code></p> <p>at the end of the document if there was a recent <code>\property_ref:nn</code> or <code>\property_ref:nnn</code> which couldn't be resolved and so raised the flag. "Recent" means in the same group or in some outer group!</p>                                                                                                                                                                                                                                                          |
| <code>\property_ref undefined_warn:n</code>  | <code>\property_ref undefined_warn:n {&lt;label&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>\property_ref undefined_warn:e</code>  | <p><b>LATEX 2<sub>E</sub>-interface:</b> not provided.</p> <p>The commands triggers the standard warning</p> <p><b>LaTeX Warning:</b> <code>There were undefined references.</code></p> <p>at the end of the document if <code>&lt;label&gt;</code> is not known. At the point where it is called it also issues the warning</p> <p><code>Reference '&lt;label&gt;' on page &lt;page&gt; undefined.</code></p>                                                                                                                                                                                                                                                                       |
| <code>\property_ref undefined_warn:nn</code> | <code>\property_ref undefined_warn:nn {&lt;label&gt;} {&lt;property&gt;}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>\property_ref undefined_warn:ee</code> | <p><b>LATEX 2<sub>E</sub>-interface:</b> see <code>\RefUndefinedWarn</code>.</p> <p>The commands triggers the standard warning</p> <p><b>LaTeX Warning:</b> <code>There were undefined references.</code></p> <p>at the end of the document if the reference can not be resolved. At the point where it is called it also issues the warning</p> <p><code>Reference '&lt;label&gt;' on page &lt;page&gt; undefined</code></p> <p>if the label is unknown, or the more specific</p> <p><code>Property '&lt;property&gt;' undefined for reference '&lt;label&gt;' on page &lt;page&gt;</code></p> <p>if the label is known but doesn't provide a value for the requested property.</p> |

---

```
\property_if_exist_p:n * \property_if_exist_p:n {\<property>}
\property_if_exist_p:e * \property_if_exist:nTF {\<property>} {\<true code>} {\<false code>}
\property_if_exist:nTF * LATEX 2 ε -interface: \IfPropertyExistsTF.
\property_if_exist:eTF * Tests if the <property> has been declared.
```

---

```
\property_if_recorded_p:n * \property_if_recorded_p:n {\<label>}
\property_if_recorded_p:e * \property_if_recorded:nTF {\<label>} {\<true code>} {\<false code>}
\property_if_recorded:nTF *
\property_if_recorded:eTF *
```

LATEX 2 $\varepsilon$ -interface: \IfLabelExistsTF  
Tests if the <label> is known. This is also true if the label has been set with the standard \label command.

---

```
\property_if_recorded_p:nn * \property_if_recorded_p:nn {\<label>} {\<property>}
\property_if_recorded_p:ee * \property_if_recorded:nnTF {\<label>} {\<property>} {\<true code>} {\<false code>}
\property_if_recorded:nnTF *
\property_if_recorded:eeTF *
```

LATEX 2 $\varepsilon$ -interface: \IfPropertyRecordedTF.  
Tests if the label <label> is known and if it provides a value of the <property>.

## 7 Auxiliary file interfaces

---

```
\new@label@record \new@label@record {\<label>} {\<data>}
```

This is a command only for use in the .aux file. It loads the key–value list of <data> to be available for the <label>.

## 8 LATEX 2 $\varepsilon$ interface

The LaTeXe interfaces always expand label and property arguments. This means that one must be careful when using active chars or commands in the names. UTF8-chars are protected and should be safe, similar most babel shorthands.

---

```
\NewProperty \NewProperty {\<property>} {\<setpoint>} {\<default>} {\<code>}
\SetProperty \SetProperty {\<property>} {\<setpoint>} {\<default>} {\<code>}
```

Sets the <property> to have the <default> specified, and at the <setpoint> (either now or shipout) to write the result of the <code> as part of a label. The <code> should be expandable. The expansion of <code> (the value of the property) is written to the .aux file and read back from there at the next compilation (at which point normally the standard LATEX catcode régime with @ a letter is active).

---

```
\RecordProperties \RecordProperties {\<label>} {\<clist>}
```

Writes the list of properties given by the <clist> to the .aux file with the <label> specified. Similar to the standard \label command the arguments are expanded. So <clist> can be a macro containing a list of properties. Also similar to the standard \label command, the command is surrounded by an \bbsphack/\esphack pair to preserve spacing.

---

`\RefProperty * \RefProperty [local default] {label} {property}`

Expands to the value of the *property* for the *label*, if available, and the default value of the property or – if given – to *local default* otherwise. If {*property*} has not been declared an error is issued.

---

`\IfPropertyExistsTF \IfPropertyExistsTF {property} {true code} {false code}`

`\IfPropertyExistsT`

`\IfPropertyExistsF`

Tests if the *property* has been declared.

---

`\IfLabelExistsTF \IfLabelExistsTF {label} {true code} {false code}`

`\IfLabelExistsT`

`\IfLabelExistsF`

Tests if the *label* has been recorded. This is also true if a label has been set with the standard `\label` command.

---

`\IfPropertyRecordedTF \IfPropertyRecordedTF {label} {property} {true code} {false code}`

`\IfPropertyRecordedT`

`\IfPropertyRecordedF`

Tests if the label and a value of the *property* for the *label* are both known.

---

`\RefUndefinedWarn \RefUndefinedWarn {label} {property}`

This command triggers the standard warning

LaTeX Warning: There were undefined references.

at the end of the document if the reference for *label* and *property* can not be resolved. At the point where it is called it also issues the warning

Reference ‘*label*’ on page *page* undefined  
if the label is unknown, or the more specific

Property ‘*property*’ undefined for reference ‘*label*’ on page *page* if  
the label is known but doesn’t provide a value for the requested property.

## 9 Pre-declared properties

---

`\abspage` (shipout) The absolute value of the current page: starts at 1 and increases monotonically at each shipout.

---

`\page` (shipout) The current page as given by `\thepage`: this may or may not be a numerical value, depending on the current style. Contrast with `\abspage`. You get this value also with the standard `\label/\pageref`.

---

`\pagenum` (shipout) The current page as arabic number. This is suitable for integer operations and comparisons.

---

`\label` (now) The content of `\@currentlabel`. This is the value that you get also with the standard `\label/\ref`.

title (now) The content of `\@currentlabelname`. This command is filled beside others by the `nameref` package and some classes (e.g. `memoir`).

target (now) The content of `\@currentHref`. This command is normally filled by for example `hyperref` and gives the name of the last destination it created.

pagetarget (shipout) The content of `\@currentHpage`. This command is filled for example by a recent version of `hyperref` and then gives the name of the last page destination it created.

counter (now) The content of `\@currentcounter`. This command contains after a `\refstepcounter` the name of the counter.

xpos (shipout) This stores the *x* and *y* coordinates of a point previously stored with ypos `\pdfsavepos/\savepos`. E.g. (if `bidi` is used it can be necessary to save the position before and after the label):

```
\tex_savepos:D
\property_record:nn{myposition}{xpos,ypos}
\tex_savepos:D
```

## 10 The Implementation

```
1 (*2ekernel | latexrelease)
2 \ExplSyntaxOn
3 (@@=property)
4 (|latexrelease|\NewModuleRelease{2023/11/01}{ltproperties}
5 (|latexrelease| {Cross-referencing-properties})
```

The approach here is based closely on that from `zref`; separate out lists of properties and the properties themselves, so the latter can be used multiple times and in varying combinations. However, not everything is a straight copy. Firstly, we treat lists of properties as simple comma lists: that allows us to have either saved or dynamic lists and to avoid another data structure. The cost is that errors are detected at point-of-use, but in any real case that should be true anyway (and is true for `\zref@labelbyprop` already). Secondly, we allow properties to have arbitrary names, as the code does not require them to tokenize as control sequences.

As properties can be reset, they are not constants. But they also have various pieces of required data. So we use the same approach as color and make them declarations. Data-wise, we need the detail of the implementation, the default and a flag to show if the code works now or at shipout. This last entry is done using text so needs a check. We could use a set of `prop` here, but as we never need to map or copy the lists, we can gain performance using the hash table approach.

```
\property_new:nnnn
\property_gset:nnnn
_property_gset:nnnn
6 \cs_new_protected:Npn \property_new:nnnn #1#2#3#4
```

```

7 {
8 \cs_if_free:cTF { __property_code_ #1 : }
9 {
10 \exp_args:Nx __property_gset:nnnn { \tl_to_str:n {#1} }
11 {#2} {#3} {#4}
12 }
13 {
14 \msg_error:nn { property }{ exists }{#1}
15 }
16 }
17 \cs_new_protected:Npn \property_gset:nnnn #1#2#3#4
18 {
19 __property_gset:ennn { \tl_to_str:n {#1} }
20 {#2} {#3} {#4}
21 }
22 \cs_new_protected:Npn __property_gset:nnnn #1#2#3#4
23 {
24 \cs_gset:cpn { __property_code_ #1 : } {#4}
25 \tl_gclear_new:c { g__property_default_ #1 _tl }
26 \tl_gset:cn { g__property_default_ #1 _tl } {#3}
27 \bool_if_exist:cF { g__property_shipout_ #1 _tl }
28 { \bool_new:c { g__property_shipout_ #1 _tl } }
29 \str_case:nnF {#2}
30 {
31 { now } { { \bool_gset_false:c { g__property_shipout_ #1 _tl } } }
32 { shipout }
33 { \bool_gset_true:c { g__property_shipout_ #1 _tl } }
34 }
35 { \msg_error:nnnn { property } { unknown-setpoint } {#1} {#2} }
36 }
37 \cs_generate_variant:Nn __property_gset:nnnn {ennn}

```

(End of definition for `\property_new:nnnn`, `\property_gset:nnnn`, and `\__property_gset:nnnn`.  
These functions are documented on page 747.)

`\NewProperty` For consistency we expand the property name, but this doesn't warrant a variant of the L3-commands.  
`\ SetProperty`

```

38 \cs_new_protected:Npn \NewProperty #1#2#3#4
39 {
40 \protected@edef\reserved@a{#1}
41 \exp_args:No \property_new:nnnn {\reserved@a} {#2}{#3}{#4}
42 }
43 \cs_new_protected:Npn \ SetProperty #1#2#3#4
44 {
45 \protected@edef\reserved@a{#1}
46 \exp_args:No \property_gset:nnnn {\reserved@a} {#2}{#3}{#4}
47 }

```

(End of definition for `\NewProperty` and `\ SetProperty`. These functions are documented on page 749.)

`\property_record:nn`  
`\property_record:nn`  
`\property_record:nnV`  
`\property_record:ee`  
`\property_record:oo`  
`\__property_record:nn`  
`\__property_record:en`  
`\__property_record_value:n`  
  `\__property_record_value_aux:n`  
  `\__property_record_value_aux:e`

Writing data when it is labelled means expanding at this stage and possibly later too. That is all pretty easy using `expl3`: we accept a stray comma at the end of the list as that is easier to deal with than trying to tidy up, and there is no real downside.

48 `\cs_new_protected:Npn \property_record:nN #1#2`

```

49 { \property_record:nV {#1} #2 }
50 \cs_new_protected:Npn \property_record:nn #1#2
51 { __property_record:en { \tl_to_str:n {#1} } {#2} }
52 \cs_generate_variant:Nn \property_record:nn { nV , ee, oo }
53 \cs_new_protected:Npn __property_record:nn #1#2
54 {
55 \protected@write \auxout {}
56 {
57 \token_to_str:N \new@label@record
58 {#1}
59 { \clist_map_function:nN {#2} __property_record_value:n }
60 }
61 }
62 \cs_generate_variant:Nn __property_record:nn { e }
63 \cs_new:Npn __property_record_value:n #1
64 { __property_record_value_aux:e { \tl_to_str:n {#1} } }
65 \cs_new:Npn __property_record_value_aux:n #1
66 {
67 \cs_if_exist:cTF { __property_code_ #1 : }
68 {
69 {#1}
70 {
71 \bool_if:cTF { g__property_shipout_ #1 _tl }
72 { \exp_not:c }
73 { \use:c }
74 { __property_code_ #1 : }
75 }
76 }
77 { \msg_expandable_error:nnn { property } { not-declared } {#1} }
78 }
79 \cs_generate_variant:Nn __property_record_value_aux:n { e }

```

(End of definition for `\property_record:nN` and others. These functions are documented on page 748.)

### \RecordProperties

```

80 \NewDocumentCommand\RecordProperties { m m }
81 {
82 \@bsphack
83 \protected@edef\reserved@a{#1}
84 \protected@edef\reserved@b{#2}
85 \property_record:oo {\reserved@a}{\reserved@b}
86 \@esphack
87 }

```

(End of definition for `\RecordProperties`. This function is documented on page 749.)

## 10.1 Reference commands

`l__property_ref_flag` A flag that is set if a reference couldn't be resolved.

```

88 \flag_new:n { l__property_ref_flag }

```

(End of definition for `l__property_ref_flag`.)

**\property\_ref:nn** Search for the label/property combination, and if not found fall back to the default of the property.

```

89 \cs_new:Npn \property_ref:nn #1#2
90 {
91 __property_ref:een
92 { \tl_to_str:n {#1} }
93 { \tl_to_str:n {#2} }
94 { \tl_use:c { g__property_default_ #2 _tl } }
95 }
96 \cs_generate_variant:Nn \property_ref:nn {ee}

```

(End of definition for **\property\_ref:nn**. This function is documented on page 748.)

**\property\_ref:nnn** This allows to set a local default value which overrides the default value of the property.

```

97 \cs_new:Npn \property_ref:nnn #1#2#3
98 {
99 __property_ref:een
100 { \tl_to_str:n {#1} }
101 { \tl_to_str:n {#2} }
102 {#3}
103 }
104 \cs_new:Npn __property_ref:nnn #1#2#3
105 {
106 \tl_if_exist:cTF { g__property_label_ #1 _ #2 _tl }
107 { \tl_use:c { g__property_label_ #1 _ #2 _tl } }
108 {
109 \flag_if_raised:nF
110 { l__property_ref_flag } { \flag_raise:n { l__property_ref_flag } }
111 }
112 }
113 \msg_expandable_error:nnn { property } { not-declared } {#2}
114 }
115 }
116 \cs_generate_variant:Nn __property_ref:nnn { ee }
117 \cs_generate_variant:Nn \property_ref:nnn {een}

```

We test for the default of the property only to check if the property has been declared.

```

111 \tl_if_exist:cTF { g__property_default_ #2 _tl }
112 { #3 }
113 { \msg_expandable_error:nnn { property } { not-declared } {#2} }
114 }
115 }
116 \cs_generate_variant:Nn __property_ref:nnn { ee }
117 \cs_generate_variant:Nn \property_ref:nnn {een}

```

(End of definition for **\property\_ref:nnn** and **\\_\_property\_ref:nnn**. This function is documented on page 748.)

**\RefProperty** Search for the label/property combination, and if not found fall back to the default of the property or the given default.

```

118 \NewExpandableDocumentCommand \RefProperty { o m m }
119 {
120 \IfNoValueTF {#1}
121 {
122 \property_ref:ee {#2}{#3}
123 }
124 {
125 \property_ref:een {#2}{#3}{#1}
126 }
127 }

```

(End of definition for **\RefProperty**. This function is documented on page 750.)

`\new@label@record` A standard recursion loop.

```

128 \cs_new_protected:Npn \new@label@record #1#2
129 {
130 \tl_if_exist:cTF { r@#1 }
131 {
132 \gdef \omultiplelabels
133 { \@latex@warning@no@line { There~were~multiply-defined~labels } }
134 \@latex@warning@no@line { Label~'#1'~multiply-defined }
135 }
136 {
137 \tl_new:c { r@#1 }
138 \tl_gset:cn { r@#1 }{#2}
139 }
140 __property_data:nnn {#1} #2 { \q_recursion_tail } { ? } \q_recursion_stop
141 }
142 \cs_new_protected:Npn __property_data:nnn #1#2#3
143 {
144 \quark_if_recursion_tail_stop:n {#2}
145 \tl_gclear_new:c { g__property_label_ \tl_to_str:n {#1} _ \tl_to_str:n {#2} _tl }
146 \tl_gset:cn { g__property_label_ \tl_to_str:n {#1} _ \tl_to_str:n {#2} _tl } {#3}
147 __property_data:nnn {#1}
148 }
```

This command is used in `\enddocument` to test if some label values have changed.

```

149 \cs_new_protected:Npn \kernel@new@label@record@testdef #1 #2
150 {
151 \tl_if_eq:cnF { r@#1 } {#2}
152 { \otempswatrue }
153 }
```

(End of definition for `\new@label@record` and `\__property_data:nnn`. This function is documented on page 749.)

## 10.2 Tests and warnings

`\property_if_exist_p:n` Tests if property has been declared.

`\property_if_exist:nTF`

```

154 \prg_new_conditional:Npnn \property_if_exist:n #1 { p , T , F , TF }
155 % #1 property
156 {
157 \cs_if_exist:cTF { __property_code_ #1 : }
158 {
159 \prg_return_true:
160 }
161 {
162 \prg_return_false:
163 }
164 }
165 \prg_generate_conditional_variant:Nnn \property_if_exist:n {e} {TF}
```

(End of definition for `\property_if_exist:nTF`. This function is documented on page 749.)

`\IfPropertyExistsTF`  
`\IfPropertyExistsT`  
`\IfPropertyExistsF`

```

166 \cs_new_eq:NN \IfPropertyExistsTF \property_if_exist:eTF
167 \cs_new:Npn \IfPropertyExistsT #1#2 {\property_if_exist:eTF {#1}{#2}{ } }
168 \cs_new:Npn \IfPropertyExistsF #1 { \property_if_exist:eTF {#1}{ } }
```

(End of definition for `\IfPropertyExistsTF`, `\IfPropertyExistsT`, and `\IfPropertyExistsF`. These functions are documented on page 750.)

`\property_if_recorded_p:n` Tests if the label has been set. This can then be used to setup e.g. rerun messages.

```

169 \prg_new_conditional:Npnn \property_if_recorded:n #1 { p , T , F , TF }
170 % #1 label
171 {
172 \tl_if_exist:cTF { r@#1 }
173 {
174 \prg_return_true:
175 }
176 {
177 \prg_return_false:
178 }
179 }
180 \prg_generate_conditional_variant:Nnn \property_if_recorded:n {e} {TF}

```

(End of definition for `\property_if_recorded:nTF`. This function is documented on page 749.)

`\IfLabelExistsTF`

```

\IfLabelExistsT
181 \cs_new_eq:NN \IfLabelExistsTF \property_if_recorded:eTF
\IfLabelExistsF
182 \cs_new:Npn \IfLabelExistsT #1#2 { \property_if_recorded:eTF {#1}{#2}{}} }
183 \cs_new:Npn \IfLabelExistsF #1 { \property_if_recorded:eTF {#1}{}} }

```

(End of definition for `\IfLabelExistsTF`, `\IfLabelExistsT`, and `\IfLabelExistsF`. These functions are documented on page 750.)

`\property_if_recorded_p:nn` tests if the label/property combination has been set. This can then be used to setup e.g. rerun messages.

```

184 \prg_new_conditional:Npnn \property_if_recorded:nn #1#2 { p , T , F , TF }
185 % #1 label #2 property
186 {
187 \tl_if_exist:cTF { g__property_label_ \tl_to_str:n {#1} _ \tl_to_str:n {#2} _tl }
188 {
189 \prg_return_true:
190 }
191 {
192 \prg_return_false:
193 }
194 }
195 \prg_generate_conditional_variant:Nnn \property_if_recorded:nn {ee} {TF}

```

(End of definition for `\property_if_recorded:nnTF`. This function is documented on page 749.)

`\IfPropertyRecordedTF`

```

\IfPropertyRecordedT
196 \cs_new_eq:NN \IfPropertyRecordedTF \property_if_recorded:eeTF
\IfPropertyRecordedF
197 \cs_new:Npn \IfPropertyRecordedT #1#2#3 { \property_if_recorded:eeTF {#1}{#2}{#3}{}} }
198 \cs_new:Npn \IfPropertyRecordedF #1#2#3 { \property_if_recorded:eeTF {#1}{#2}{}}{#3} }

```

(End of definition for `\IfPropertyRecordedTF`, `\IfPropertyRecordedT`, and `\IfPropertyRecordedF`. These functions are documented on page 750.)

```
\property_ref undefined_warn: \G@refundefinedtrue is defined in ltxref and redefines a warning message.
```

```
199 \cs_new_protected:Npn \property_ref undefined_warn:
200 {
201 \flag_if_raised:nT { l__property_ref_flag }
202 {
203 \G@refundefinedtrue
204 }
205 }
```

(End of definition for `\property_ref undefined_warn:`. This function is documented on page 748.)

```
\property_ref undefined_warn:n
```

```
206 \cs_new_protected:Npn \property_ref undefined_warn:n #1 %#1 label
207 {
208 \property_if_recorded:nF {#1}
209 {
210 \G@refundefinedtrue
211 \@latex@warning{Reference`#1'~on-page~\thepage\space undefined}%
212 }
213 }
```

(End of definition for `\property_ref undefined_warn:n`. This function is documented on page 748.)

```
\property_ref undefined_warn:nn
```

```
\property_ref undefined_warn:ee
```

```
\RefUndefinedWarn
214 \cs_new_protected:Npn \property_ref undefined_warn:nn #1#2 %#1 label, #2 property
215 {
216 \property_if_recorded:nTF {#1}
217 {
218 \property_if_recorded:nnF {#1}{#2}
219 {
220 \G@refundefinedtrue
221 \@latex@warning
222 { Property`#2'~undefined~for~reference`#1'~on~page~\thepage }
223 }
224 }
225 {
226 \G@refundefinedtrue
227 \@latex@warning { Reference`#1'~on~page~\thepage\space undefined }%
228 }
229 }
230 \cs_generate_variant:Nn \property_ref undefined_warn:nn {ee}
231 \cs_set_eq:NN \RefUndefinedWarn \property_ref undefined_warn:ee
```

(End of definition for `\property_ref undefined_warn:nn` and `\RefUndefinedWarn`. These functions are documented on page 748.)

### 10.3 Predeclared properties

```
abspage
```

```
232 \property_new:nnnn { abspage } { shipout }
233 { 0 } { \int_use:N \g_shipout_READONLY_int }
```

(End of definition for `abspage`. This variable is documented on page 750.)

**page**

234 \property\_new:nnnn { page } { shipout } { 0 } { \thepage }

*(End of definition for page. This variable is documented on page 750.)*

**pagenum**

235 \property\_new:nnnn { pagenum } { shipout } { 0 } { \the \value { page } }

*(End of definition for pagenum. This variable is documented on page 750.)*

**label**

236 \property\_new:nnnn { label } { now } { ?? } { \@currentlabel }

*(End of definition for label. This variable is documented on page 750.)*

**title**

237 \property\_new:nnnn { title } { now }

238 { \exp\_not:n { \textbf { ?? } } } { \@currentlabelname }

*(End of definition for title. This variable is documented on page 751.)*

**target**

239 \property\_new:nnnn { target } { now } { } { \@currentHref }

*(End of definition for target. This variable is documented on page 751.)*

**target**

240 \newcommand{\@currentHpage{}}

241 \property\_new:nnnn { pagetarget } { shipout } { } { \@currentHpage }

*(End of definition for target. This variable is documented on page 751.)*

**counter**

242 \property\_new:nnnn { counter } { now } { } { \@currentcounter }

*(End of definition for counter. This variable is documented on page 751.)*

**xpos**

**ypos** 243 \property\_new:nnnn { xpos } { shipout } { 0 } { \int\_use:N \tex\_lastxpos:D }

244 \property\_new:nnnn { ypos } { shipout } { 0 } { \int\_use:N \tex\_lastypos:D }

*(End of definition for xpos and ypos. These variables are documented on page 751.)*

## 10.4 Messages

```
245 \msg_new:nnnn { property } { exists }
246 { Property~'#1'~ has~ already~ been~ declared. }
247 { There~ already~ exists~ a~ property~ declaration~ with~ this~
248 name.\\
249 Please~ use~ a~ different~ name~ for~ your~ property.}
250
251 \msg_new:nnnn { property } { not-declared }
252 { Property~'#1'~not-declared. }
253 {
254 LaTeX-has~been~asked~to~use~property~'#1',~but~this~
255 name~has~not~been~declared.
256 }
257 \msg_new:nnnn { property } { unknown-setpoint }
258 { Unknown~keyword~'#2'~for~setting~property~'#1'. }
259 {
260 LaTeX-has~been~asked~to~set~the~property~'#1',~but~the~keyword~
261 '#2'~is~not~one~of~the~two~known~values:~'now'~or~'shipout'.
262 }
263 %
264 <|latexrelease>\IncludeInRelease{0000/00/00}{ltproperties}
265 <|latexrelease> {cross-referencing-properties~(undo)}%
266 <|latexrelease>
267 <|latexrelease>\let \NewProperty \@undefined
268 <|latexrelease>\let \SetProperty \@undefined
269 <|latexrelease>
270 <|latexrelease>\let \RecordProperties \@undefined
271 <|latexrelease>\let \RefProperty \@undefined
272 <|latexrelease>\let \RefUndefinedWarn \@undefined
273 <|latexrelease>
274 <|latexrelease>\let \IfPropertyExistsTF \@undefined
275 <|latexrelease>\let \IfLabelExistsTF \@undefined
276 <|latexrelease>\let \IfPropertyRecordedTF \@undefined
277 <|latexrelease>
278 <|latexrelease>\let\new@label@record \@undefined
279 <|latexrelease>\let\@kernel@new@label@record@testdef\@undefined
280 <|latexrelease>\EndModuleRelease
281 \ExplSyntaxOff
282 </2ekernel | latexrelease>
 Reset module prefix:
283 <@@=>
```

# File 37

## 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`.

```
1 <*2ekernel>
2 \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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
\enddocument ==
BEGIN
 \@checkend{document} %% checks for unmatched \begin
 \clearpage
 \begingroup
 if @filesw = true
 then close file @mainaux
 if G@refundefined = true
 then LaTeX Warning: 'There are undefined references.' fi
 if @multiplelabels = true
 then LaTeX Warning:
 'One or more label(s) multiply defined.'
 else
 \@setckpt {ARG1}{ARG2} == null
 \newlabel{LABEL}{VAL} ==
 BEGIN
 \reserved@a == VAL
 if def(\reserved@a) = def(\r@LABEL)
 else @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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

\@currenvir The name of the current environment. Initialized to document to so that \end{document} works correctly.

```

3 \def\@currenvir{document}
```

*(End of definition for \@currenvir.)*

```

\if@ignore
\@ignorettrue
\@ignorefalse
4 \def\@ignorefalse{\global\let\if@ignore\iffalse}
5 \def\@ignorettrue {\global\let\if@ignore\iftrue}
6 \@ignorefalse

```

*(End of definition for \if@ignore, \@ignorettrue, and \@ignorefalse.)*

\ignorespacesafterend

```

7 \let\ignorespacesafterend\@ignorettrue
```

*(End of definition for \ignorespacesafterend.)*

\end{document})

```

8 </2ekernel>
9 {*2ekernel | latexrelease}
10 {latexrelease}\IncludeInRelease{2023/11/01}%
11 {latexrelease} {\enddocument}{check property labels}%
12 \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...

```

13 \@kernel@before@enddocument
14 \UseOneTimeHook{enddocument}%
15 \@kernel@after@enddocument
```

```

16 \@checkend{document}%
17 \clearpage
18 \UseOneTimeHook{enddocument/afterlastpage}%
19 \@kernel@after@enddocument@afterlastpage
20 \begingroup
21 \if@filesw
22 \immediate\closeout\@mainaux
23 \let\@setckpt\@gobbletwo
24 \let\@newl@bel\@testdef
25 \let\newlabel@record\@kernel@newlabel@record@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`.

```

26 \@tempswafalse
27 \makeatletter \@@input\jobname.aux
28 \fi
29 \UseOneTimeHook{enddocument/afteraux}%

```

Next hook is expect to contain only code for writing info messages on the terminal.

```

30 \UseOneTimeHook{enddocument/info}%
31 \endgroup
32 \UseOneTimeHook{enddocument/end}%
33 \deadcycles\z@\@@end}

```

The public hooks used in `\enddocument`:

```

34 \NewHook{enddocument}
35 \NewHook{enddocument/afterlastpage}
36 \NewHook{enddocument/afteraux}
37 \NewHook{enddocument/info}
38 \NewHook{enddocument/end}

```

This is one of the few places where we already add data and rules to a hook already in the kernel.

If we roll back we have to drop stuff before adding chunks, otherwise the code will just be appended, and thus doubled. This would result in a harmless warning during the format generation, because in that case the code chunk label doesn't exist, and therefore can't be dropped.

```

39 <|latexrelease>\RemoveFromHook{enddocument/info}[kernel/filelist]
40 <|latexrelease>\RemoveFromHook{enddocument/info}[kernel/warnings]
41 <|latexrelease>\RemoveFromHook{enddocument/info}[kernel/release]
42 \AddToHook{enddocument/info}[kernel/filelist]{\@dofilelist}
43 \AddToHook{enddocument/info}[kernel/warnings]{\@enddocument@kernel@warnings}
44 \AddToHook{enddocument/info}[kernel/release]{%
45 \let\show@release@info\wlog
46 \show@release@info{ ****}%
47 \the\LaTeXReleaseInfo
48 \show@release@info{ ****}%
49
50 \DeclareHookRule{enddocument/info}{kernel/release}{before}{kernel/filelist}
51 \DeclareHookRule{enddocument/info}{kernel/filelist}{before}{kernel/warnings}

```

(End of definition for \enddocument.)

```
\@enddocument@kernel@warnings
```

```
52 \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.

```
53 \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.

```
54 \@font@warning{Size substitutions with differences\MessageBreak
55 up to \font@submax\space have occurred.\gobbletwo}%
56 \fi
```

The macro \defaultsubs is initially \relax but gets redefined to produce a warning if there have been some default font substitutions.

```
57 \defaultsubs
```

The macro \refundefined is initially \relax but gets redefined to produce a warning if there are undefined refs.

```
58 \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 variorref that generates labels on the fly), we suppress this message.

```
59 \if@filesw
60 \ifx \multiplelabels \relax
61 \if@tempswa
62 \@latex@warning{no@line{Label(s) may have changed.
63 Rerun to get cross-references right}%
64 \fi
65 \else
66 \multiplelabels
67 \fi
68 \ifx \extra@page@added \relax
69 \@latex@warning{no@line{Temporary extra page added at the end.
70 Rerun to get it removed}%
71 \fi
72 \fi
73 }
```

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.

```
72 \fi
73 }
```

(End of definition for \enddocument@kernel@warnings.)

```
74 </2ekernel | latexrelease>
75 <| latexrelease>\EndIncludeInRelease
76 <| latexrelease>\IncludeInRelease{2020/10/01}%
77 <| latexrelease> {\enddocument}{Use Hooks}%
78 <| latexrelease>\def\enddocument{%
79 <| latexrelease> \@kernel@before@enddocument
80 <| latexrelease> \UseOneTimeHook{enddocument}%
81 <| latexrelease> \@kernel@aftersenddocument
```

```
82 <latextitle> \@checkend{document}%
83 <latextitle> \clearpage
84 <latextitle> \UseOneTimeHook{enddocument/afterlastpage}%
85 <latextitle> \@kernel@after@enddocument@afterlastpage
86 <latextitle> \begingroup
87 <latextitle> \if@filesw
88 <latextitle> \immediate\closeout\@mainaux
89 <latextitle> \let\@setckpt\@gobbletwo
90 <latextitle> \let\@newl@bel\@testdef
91 <latextitle> \tempswafalse
92 <latextitle> \makeatletter \@@input\jobname.aux
93 <latextitle> \fi
94 <latextitle> \UseOneTimeHook{enddocument/afteraux}%
95 <latextitle> \UseOneTimeHook{enddocument/info}%
96 <latextitle> \endgroup
97 <latextitle> \UseOneTimeHook{enddocument/end}%
98 <latextitle> \deadcycles{z@\@@end}
99 <latextitle> \NewHook{enddocument}
100 <latextitle> \NewHook{enddocument/afterlastpage}
101 <latextitle> \NewHook{enddocument/afteraux}%
102 <latextitle> \NewHook{enddocument/info}%
103 <latextitle> \NewHook{enddocument/end}
```

if we roll back we have to drop stuff before adding chunks, otherwise the code will just be appended, and thus doubled.

```
104 <{latexrelease}\RemoveFromHook{enddocument/info}[kernel/filelist]
105 <{latexrelease}\RemoveFromHook{enddocument/info}[kernel/warnings]
106 <{latexrelease}\RemoveFromHook{enddocument/info}[kernel/release]

107 <{latexrelease}\AddToHook{enddocument/info}[kernel/filelist]{\@dofilelist}
108 <{latexrelease}\AddToHook{enddocument/info}[kernel/warnings]{\@enddocument@kernel@warnings}
109 <{latexrelease}\AddToHook{enddocument/info}[kernel/release]{%
110 \let\show@release@info\wlog
111 \show@release@info{ **** * * * * }%
112 \the\LaTeXReleaseInfo
113 \show@release@info{ **** * * * * }%
114 }

115 <{latexrelease}\DeclareHookRule{enddocument/info}[kernel/release]{before}{kernel/filelist}
116 <{latexrelease}\DeclareHookRule{enddocument/info}[kernel/filelist]{before}{kernel/warnings}
117 <{latexrelease}\def\@enddocument@kernel@warnings{%
118 \ifdim \font@submax >\fontsubfuzz\relax
119 \font@warning{Size substitutions with differences\MessageBreak
120 up to \font@submax\space have occurred.\@gobbletwo}%
121 \fi
122 \font@defaultsubs
123 \font@undefined
124 \if@filesw
125 \ifx \multiplelabels \relax
126 \if@tempswa
127 \font@warning{no@line{Label(s)} may have changed.
128 Rerun to get cross-references right}%
129 \fi
130 \else
131 \multiplelabels
132 \fi
133 }
```

```

133 <|latexrelease> \ifx \c@extra@page@added \relax
134 <|latexrelease> \@latex@warning@no@line{Temporary extra page added at the end.
135 <|latexrelease> Rerun to get it removed}%
136 <|latexrelease> \fi
137 <|latexrelease> \fi
138 <|latexrelease> }
139 <|latexrelease>\EndIncludeInRelease
140 <|latexrelease>\IncludeInRelease{0000/00/00}%
141 <|latexrelease> {\enddocument}{Use Hooks}%
142 <|latexrelease>
143 <|latexrelease>\def\enddocument{%
144 <|latexrelease> \let\AtEndDocument\c@firstofone
145 <|latexrelease> \c@enddocumenthook
146 <|latexrelease> \c@checkend{document}%
147 <|latexrelease> \clearpage
148 <|latexrelease> \begingroup
149 <|latexrelease> \if@filesw
150 <|latexrelease> \immediate\closeout\c@mainaux
151 <|latexrelease> \let\c@setckpt\c@gobbletwo
152 <|latexrelease> \let\c@newl@bel\c@testdef
153 <|latexrelease> \c@tempswafalse
154 <|latexrelease> \makeatletter \c@input\jobname.aux
155 <|latexrelease> \fi
156 <|latexrelease> \c@odfichierlist
157 <|latexrelease> \ifdim \font@submax >\fontsubfuzz\relax
158 <|latexrelease> \c@font@warning{Size substitutions with differences\MessageBreak
159 <|latexrelease> up to \font@submax\space have occurred.\c@gobbletwo}%
160 <|latexrelease> \fi
161 <|latexrelease> \c@defaultsubs
162 <|latexrelease> \c@refundefined
163 <|latexrelease> \if@filesw
164 <|latexrelease> \ifx \c@multipletlabels \relax
165 <|latexrelease> \if@tempswa
166 <|latexrelease> \@latex@warning@no@line{Label(s) may have changed.
167 <|latexrelease> Rerun to get cross-references right}%
168 <|latexrelease> \fi
169 <|latexrelease> \else
170 <|latexrelease> \c@multipletlabels
171 <|latexrelease> \fi
172 <|latexrelease> \fi
173 <|latexrelease> \endgroup
174 <|latexrelease> \deadcycles\z@\c@end}
175 <|latexrelease>
176 <|latexrelease>\let\c@enddocument\c@kernel@c@warnings\c@undefined
177 <|latexrelease>
178 <|latexrelease>\EndIncludeInRelease
179 <|2ekernel>

```

#### \c@kernel@c@before@cenddocument

The \c@kernel@c@before@cenddocument hook is slightly different because we initialize it with \par so that \enddocument always returns to vertical mode as its first action.

```

180 </2ekernel>
181 <|2ekernel | latexrelease>
182 <|latexrelease>\IncludeInRelease{2021/06/01}%
183 <|latexrelease> {\c@kernel@c@before@cenddocument}{kernel before hook}%

```

```

184 \def\@kernel@before@enddocument{\par}
185 </2ekernel | latexrelease>
186 \<latexrelease>\EndIncludeInRelease
 The rollback code renders it harmless.
187 \<latexrelease>\IncludeInRelease{0000/00/00}%
188 \<latexrelease> {\@kernel@before@enddocument}{kernel before hook}%
189 \<latexrelease>
190 \<latexrelease>\let\@kernel@before@enddocument\empty
191 \<latexrelease>
192 \<latexrelease>\EndIncludeInRelease
193 <*2ekernel>

```

(*End of definition for \@kernel@before@enddocument.*)

\@testdef

```

194 \def\@testdef #1#2#3{%
195 \def\reserved@a{#3}\expandafter \ifx \csname #1#2\endcsname
196 \reserved@a \else \tempswatru \fi}

```

(*End of definition for \@testdef.*)

Reading data from auxiliary files (like `.toc` normally happens in vertical mode and it therefore doesn't matter if line endings are converted to spaces by TeX during that process.

However, especially the `.toc` file might be read in L-R mode (in cases the `\tableofcontents` attempts to put, say, a list of sub-sections as a paragraph). In that case the newlines after a line like

```
\contentsline {subsubsection}{\numberline {1.1.1}A C-head}{2}
```

might result in spurious spaces (e.g., when that level is not included).

That could be fixed by reading in the file using `\endlinechar=-1` but that has the danger that it drops some valid endlines that should be converted to spaces (for example, when the user edited the TOC and then used `\nofiles` to preserve it).

So the approach taken instead is this:

- `\addcontentsline` adds the command `\protected@file@percent` to the end of the second argument of `\@writefile` that is written to the `.aux`. As the name indicates this is a protected macro so it doesn't change if it is written out.
- When the `.aux` is read back in at the end of the run, `\@writefile` is executed and writes its second argument unmodified to the file with the extension given by its first argument. Or rather that was how it was in the past.
- Instead we change `\@writefile` slightly: basically it looks at the second argument and if the last token in there is `\protected@file@percent` then it is replaced by a percent character and that is then written out. If not (for example, if the data came from a user issued `\addtocontents`, or from some package that uses `\@writefile` for writing its own files) then the command behaves exactly as before.

`\protected@file@percent` Dummy cs to be replaced by a percent sign inside `\@writefile`. If it survives (when used incorrectly) it will expand to nothing in a typesetting context.

```

197 </2ekernel>
198 <*2ekernel | latexrelease>

```

```

199 \iflatexrelease\IncludeInRelease{2018/12/01}%
200 \iflatexrelease
201 {\protected@file@percent}{Mask line endings}%

```

(End of definition for \protected@file@percent.)

\add@percent@to@temptokena Helper function which is used to inspect a sequence of tokens (the second argument of \writefile and if the last token is \protected@file@percent it will replace it by a harmless percent. The result is saved in \temptokena for later use.

```

202 \catcode`\^^A=9
203 \long\gdef\add@percent@to@temptokena
204 #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.

```

205 {\expandafter\ifx\expandafter X\detokenize{#2}X\expandafter\dont@add@percent@to@temptokena
206 \expandafter\do@add@percent@to@temptokena\fi{#1}}
207 \long\def\dont@add@percent@to@temptokena#1{%
208 \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 :-)

```

209 \begingroup
210 \catcode`\%=12
211 \catcode`\^^A=14
212 \long\gdef\do@add@percent@to@temptokena#1{\temptokena\expandafter{#1}%``A

```

Can't be on the same line as the `%` — see above.

```

213 }
214 \endgroup

```

(End of definition for \add@percent@to@temptokena.)

\@writefile

```

215 \long\def\@writefile#1#2{%
216 \@ifundefined{tf@#1}\relax
217 {%

```

If we write to the file we first prepare #2 using \add@percent@to@temptokena and then write the token register out.

```

218 \add@percent@to@temptokena
219 \empty#2\protected@file@percent
220 \add@percent@to@temptokena
221 \immediate\write\csname tf@#1\endcsname{\the\temptokena}%
222 }%
223 }

```

```

224 </2ekernel | latexrelease>
225 <latexrelease>\EndIncludeInRelease
226 <latexrelease>\IncludeInRelease{0000/00/00}%
227 <latexrelease> {\protected@file@percent}{Mask line endings}%
228 <latexrelease>\let\protected@file@percent\@undefined
229 <latexrelease>\let\add@percent@to@temptokena\@undefined
230 <latexrelease>\let\do@add@percent@to@temptokena\@undefined
231 <latexrelease>\let\dont@add@percent@to@temptokena\@undefined
232 <latexrelease>\long\def\@writefile#1#2{%
233 <latexrelease> \@ifundefined{tf@#1}\relax
234 <latexrelease> {\@temptokena{#2}}%
235 <latexrelease> \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
236 <latexrelease> }%
237 <latexrelease>}
238 <latexrelease>\EndIncludeInRelease
239 <*2ekernel>

```

(End of definition for `\@writefile`.)

`\stop`

```

240 \def\stop{\clearpage\deadcycles\z@\let\par\@@par\@@end}

```

(End of definition for `\stop`.)

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```

241 \everypar{\@nodocument} %% To get an error if text appears before the
242 \nullfont %% \begin{document}

```

`\begin`, `\end`, and `\@checkend` changed so `\end{document}` will catch an unmatched `\begin`. Changed 24 May 89 as suggested by Frank Mittelbach and Rainer Sch\"opf.

```

\begin{NAME} ==
BEGIN
 IF \NAME undefined THEN \reserved@a == BEGIN report error END
 ELSE \reserved@a ==
 (@currenvir :=L NAME) \NAME
 FI
 @ignore :=G F %% Added 30 Nov 88
 \begingroup
 \@endpe := F
 \@currenvir :=L NAME
 \NAME
END

\end{NAME} ==
BEGIN
 \endNAME
 \@checkend{NAME}
 \endgroup
 IF \@endpe = T %% @endpe set True by \@endparenv

```

```

THEN \@doendpe %% \@doendpe redefines \par and \everypar
%% to suppress paragraph indentation in
%% immediately following text
FI
IF @ignore = T
THEN @ignore :=G F
 \ignorespaces
FI
END

\@checkend{NAME} ==
BEGIN
IF \@currenvir = NAME
ELSE \@badend{NAME}
FI
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```
\begin{code}
243 </2ekernel>
244 <*2ekernel | latexrelease>
245 <latexrelease>\IncludeInRelease{2020/10/01}%
246 <latexrelease> {\begin}{Use hook system}%
247 \protected\def\begin#1{%
248 \UseHook{env/#1/before}%
249 \@ifundefined{#1}%
250 {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
251 {\def\reserved@a{\def\@currenvir{#1}%
252 \edef\@currenvline{\on@line}%
253 \@execute@begin@hook{#1}%
254 \csname #1\endcsname}%
255 \ignorespace
256 \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 been found all this testing is no longer needed and so we redefine \@execute@begin@hook to simply use the hook.

```

257 \def\@execute@begin@hook #1{%
258 \expandafter\ifx\csname #1\endcsname\document
259 \endgroup
260 \gdef\@execute@begin@hook##1{\UseHook{env/##1/begin}}%
261 \@expl@@@initialize@call@o
262 \fi

```

If this is an environment before \begin{document} we just run the hook so this can be outside the test.

```

263 \UseHook{env/#1/begin}%
264 }
265 </2ekernel | latexrelease>
266 <latexrelease>\EndIncludeInRelease

```

```

267 〈latexrelease〉\IncludeInRelease{2019/10/01}%
268 〈latexrelease〉 {\begin}{Making \begin/\end robust}%
269 〈latexrelease〉\DeclareRobustCommand\begin[1]{%
270 〈latexrelease〉 \@ifundefined{#1}%
271 〈latexrelease〉 {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
272 〈latexrelease〉 {\def\reserved@a{\def\@currenvir{#1}}%
273 〈latexrelease〉 \edef\@currenvline{\on@line}%
274 〈latexrelease〉 \csname #1\endcsname}%
275 〈latexrelease〉 \@ignorefalse
276 〈latexrelease〉 \begingroup\@endpefalse\reserved@a}
277 〈latexrelease〉\EndIncludeInRelease

```

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.

```

278 \% \edef\begin
279 \% {\unexpanded{%
280 \% \ifx\protect\@typeset@protect
281 \% \expandafter\@gobble
282 \% \fi
283 \% \protect
284 \% }%
285 \% \expandafter\noexpand\csname begin \endcsname
286 \% }
287 \% \namedef{begin }#1{%
288 \% \@ifundefined{#1}%
289 \% {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
290 \% {\def\reserved@a{\def\@currenvir{#1}}%
291 \% \edef\@currenvline{\on@line}%
292 \% \csname #1\endcsname}%
293 \% \@ignorefalse
294 \% \begingroup\@endpefalse\reserved@a}
295 〈latexrelease〉\IncludeInRelease{0000/00/00}%
296 〈latexrelease〉 {\begin}{Making \begin/\end robust}%
297 〈latexrelease〉\def\begin#1{%
298 〈latexrelease〉 \@ifundefined{#1}%
299 〈latexrelease〉 {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
300 〈latexrelease〉 {\def\reserved@a{\def\@currenvir{#1}}%
301 〈latexrelease〉 \edef\@currenvline{\on@line}%
302 〈latexrelease〉 \csname #1\endcsname}%
303 〈latexrelease〉 \@ignorefalse
304 〈latexrelease〉 \begingroup\@endpefalse\reserved@a}
305 〈latexrelease〉

```

Also undo the internal commands as some packages unfortunately test for their existence instead of using `\IfFormatAtLeastTF`.

```

306 〈latexrelease〉\expandafter\let\csname begin \endcsname\@undefined
307 〈latexrelease〉
308 〈latexrelease〉\EndIncludeInRelease
309 {*2ekernel}

```

*(End of definition for `\begin`.)*

`\end` The top level definition for `\end`.

```

310 </2ekernel>
311 <*2ekernel | latexrelease>
312 <latexrelease>\IncludeInRelease{2019/10/01}%
313 <latexrelease> {\end}{Making \begin/\end robust}%

```

While `\begin` was made robust simply by using `\DeclareRobustCommand` we need to be a bit more subtle with `\end` as there are packages out there that try to look into the top-level contents of `\end{foo}` (that is at the expansion of `\endfoo`) to see if it contains certain macros. This is done by hitting `\end{foo}` with three `\expandafters`, the first to get

```
\csname endfoo\endcsname @checkend{foo}% etc.
```

the second to expand the `\csname`, i.e., to get to

```
\endfoo @checkend{foo}% etc.
```

and the third to finally get to the top-level content of `\endfoo`, i.e.

```
<top-level content of \endfoo> @checkend{foo}% etc.
```

Therefore a robust replacement should produce the same results after three expansions (there first is obviously different).

Basically the definition of `\end` should either produce `\protect\end_` (when not doing typesetting) or it should produce `\end_` (without the `\protect`) when doing typesetting. Furthermore, it should (when in typesetting mode) show exactly the same result as `\end_` (which is the original fragile definition of `\end`) when you expand either of them twice, i.e.,

```
\endfoo @checkend{foo}% etc.
```

That is achieved with the code below (which is worth studying carefully).

There is some trickery involved here: in particular we use `\romannumeral` to change a single expansion into three successive expansions in one go. That primitive expands until it has scanned a number (0 in this case, so it doesn't produce any output) and so it allows us to place arbitrary many `\expandafters` inside that are all going to be executed when `\romannumeral` is hit by a single `\expandafter`.

```

314 \edef\end
315 {\unexpanded{%
316 \romannumeral
317 \ifx\protect\@typeset@protect
318 \expandafter%1
319 \expandafter%2
320 \expandafter%1
321 \expandafter%1 %3 expands the \csname inside \end<space>
322 \expandafter%1
323 \expandafter%2 %2 expands \end<space>
324 \expandafter%1 %1 expands the \else
325 \z@
326 \else
327 \expandafter\z@\expandafter\protect
328 \fi
329 }%
330 \expandafter\noexpand\csname end \endcsname
331 }
332 </2ekernel | latexrelease>
333 <latexrelease>\EndIncludeInRelease

```

And here is the original definition of `\end` the way it was in L<sup>A</sup>T<sub>E</sub>X for several decades now hidden in `\end_`.

```

334 <latexrelease>\IncludeInRelease{0000/00/00}%
335 <latexrelease> {\end}{Making \begin/\end robust}%
336 <latexrelease>\def\end#1{%
337 <latexrelease> \csname end#1\endcsname\@checkend{#1}%
338 <latexrelease> \expandafter\endgroup\if@endpe\@doendpe\fi
339 <latexrelease> \if@ignore\@ignorefalse\ignorespaces\fi}
340 <latexrelease>
341 <latexrelease>\EndIncludeInRelease
342 <*2ekernel>
```

*(End of definition for `\end`.)*

`\end\verbvisible` The internal version with a space at the end.

```

343 </2ekernel>
344 <*2ekernel | latexrelease>
345 <latexrelease>\IncludeInRelease{2024/11/01}%
346 <latexrelease> {\end!space}{New \endpe handling}%
347 \@namedef{end }#1{%
348 \romannumeral
349 \IfHookEmptyTF{env/#1/end}%
350 {\expandafter\z@}%
351 {\z@\UseHook{env/#1/end}}%
352 \csname end#1\endcsname\@checkend{#1}%
353 }
```

We can now close the environment group and due to the new `\if@endpe` handling we no longer need to `\expandafter` out of the group.

```

353 % \expandafter\endgroup\if@endpe\@doendpe\fi
354 \endgroup
355 \UseHook{env/#1/after}%
356 \if@ignore\@ignorefalse\ignorespaces\fi
357 }
358 </2ekernel | latexrelease>
359 <latexrelease>\EndIncludeInRelease
```

Version that adds hooks (so different from the 2019 version). It fixes tlb3722 but the change should perhaps be made in `tabularx` instead.

```

360 % \begin{macrocode}
361 <latexrelease>\IncludeInRelease{2020/10/01}%
362 <latexrelease> {\end!space}{Use hook system}%
363 <latexrelease>
364 <latexrelease>\@namedef{end }#1{%
365 \romannumeral
366 \IfHookEmptyTF{env/#1/end}%
367 {\expandafter\z@}%
368 {\z@\UseHook{env/#1/end}}%
369 \csname end#1\endcsname\@checkend{#1}%
370 \expandafter\endgroup\if@endpe\@doendpe\fi
371 \UseHook{env/#1/after}%
372 \if@ignore\@ignorefalse\ignorespaces\fi
373 }
374 <latexrelease>\EndIncludeInRelease
```

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}%
 375 〈latexrelease〉\IncludeInRelease{2019/10/01}%
 376 〈latexrelease〉 {\end!space}{Making \begin{/end robust}%
 377 〈latexrelease〉
 378 〈latexrelease〉\@namedef{end }#1{%
 379 〈latexrelease〉 \csname end#1\endcsname\@checkend{#1}%
 380 〈latexrelease〉 \expandafter\endgroup\if@endpe\@doendpe\fi
 381 〈latexrelease〉 \if@ignore\@ignorefalse\ignorespaces\fi}%
 382 〈latexrelease〉\EndIncludeInRelease
 383 〈latexrelease〉\IncludeInRelease{0000/00/00}%
 384 〈latexrelease〉 {\end!space}{Making \begin{/end robust}%

```

Undo the internal command as some packages unfortunately test for their existence instead of using `\IfFormatAtLeastTF`.

```

 385 〈latexrelease〉\expandafter\let\csname end \endcsname\@undefined
 386 〈latexrelease〉
 387 〈latexrelease〉\EndIncludeInRelease
 388 〈*2ekernel〉

```

*(End of definition for `\end\verbvisiblespace`.)*

```

\@checkend
 389 \def\@checkend#1{\def\reserved@a{#1}\ifx
 390 \reserved@a\@currenvir \else\@badend{#1}\fi}

```

*(End of definition for `\@checkend`.)*

**\@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.

```

 391 \let\@currenvline\@empty

```

*(End of definition for `\@currenvline`.)*

**\AtBeginEnvironment**    **\AtEndEnvironment**  
**\BeforeBeginEnvironment**    **\AfterEndEnvironment** We provide 4 high-level hook interfaces directly, the others only when etoolbox is loaded

```

 392 〈/2ekernel〉
 393 〈*2ekernel | latexrelease〉
 394 〈latexrelease〉\IncludeInRelease{2020/10/01}%
 395 〈latexrelease〉 {\AtBeginEnvironment}{Hooks for environments}%
 396 \newcommand\AtBeginEnvironment[2][]{\AddToHook{env/#2/begin}{#1}}
 397 \newcommand\AtEndEnvironment[2][]{\AddToHook{env/#2/end}{#1}}
 398 \newcommand\BeforeBeginEnvironment[2][]{\AddToHook{env/#2/before}{#1}}
 399 \newcommand\AfterEndEnvironment[2][]{\AddToHook{env/#2/after}{#1}}
 400 〈/2ekernel | latexrelease〉
 401 〈latexrelease〉\EndIncludeInRelease

```

```

402 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
403 ⟨latexrelease⟩ {\AtBeginEnvironment}{Hooks for environments}%
404 ⟨latexrelease⟩
405 ⟨latexrelease⟩\let\AtBeginEnvironment\@undefined
406 ⟨latexrelease⟩\let\AtEndEnvironment\@undefined
407 ⟨latexrelease⟩\let\BeforeBeginEnvironment\@undefined
408 ⟨latexrelease⟩\let\AfterEndEnvironment\@undefined
409 ⟨latexrelease⟩
410 ⟨latexrelease⟩\EndIncludeInRelease
411 ⟨*2ekernel⟩

```

(End of definition for `\AtBeginEnvironment` and others. These functions are documented on page [218](#).)

## 1.2 Center, Flushright, Flushleft

```
412 \message{center,}
```

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```

\center, \flushright and \flushleft set
\rightskip = 0pt or \@flushglue (as appropriate)
\leftskip = 0pt or \@flushglue (as appropriate)
\parindent = 0pt
\parfillskip = 0pt. (except \flushleft)
\\ == \par \vskip -\parskip
\\[LENGTH] == \\ \vskip LENGTH
* == \par \penalty 10000 \vskip -\parskip
*[LEN] == * \vskip LENGTH

```

They invoke the trivlist environment to handle vertical spacing before and after them.

`\centering`, `\raggedright` and `\raggedleft` are the declaration analogs of the above.

`\raggedright` has a more universal effect, however. It sets `\rightskip := flushglue`. Every environment, like the list environments, that set `\rightskip` to its 'normal' value set it to `\@rightskip`

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```
\@centercr
```

```

413 ⟨/2ekernel⟩
414 ⟨*2ekernel | latexrelease⟩
415 ⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
416 ⟨latexrelease⟩ {\@centercr}{Make robust}%
417 \protected\def\@centercr{\ifhmode \unskip\else \nolnerr\fi
418 \par\@ifstar{\nobreak\@centercr}\@xcentercr}
419 ⟨/2ekernel | latexrelease⟩

```

```

420 〈\latexrelease〉\EndIncludeInRelease
421 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
422 〈\latexrelease〉 {\@centercr}{\Make robust}%
423 〈\latexrelease〉
424 〈\latexrelease〉\def\@centercr{\ifhmode \unskip\else \nolnerr\fi
425 〈\latexrelease〉 \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
426 〈\latexrelease〉
427 〈\latexrelease〉\EndIncludeInRelease
428 (*2ekernel)

```

(End of definition for \@centercr.)

\@xcentercr

```

429 \def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
430 [\\@icentercr\ignorespaces}

```

(End of definition for \@xcentercr.)

\@icentercr

```

431 〈/2ekernel〉
432 〈*2ekernel | \latexrelease〉
433 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
434 〈\latexrelease〉 {\@icentercr}{centering, etc support calc}%
435 \def\@icentercr[#1]{\@vspace@calcify{#1}\ignorespaces}
436 〈/2ekernel | \latexrelease〉
437 〈\latexrelease〉\EndIncludeInRelease
438 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
439 〈\latexrelease〉 {\@icentercr}{centering, etc support calc}%
440 〈\latexrelease〉
441 〈\latexrelease〉\def\@icentercr[#1]{\vskip #1\ignorespaces}
442 〈\latexrelease〉\EndIncludeInRelease
443 (*2ekernel)

```

(End of definition for \@icentercr.)

**center** (env.) We use \relax to prevent \item scanning too far.

```

444 \def\centerf{\trivlist \centering\item\relax}
445 \def\endcenter{\endtrivlist}
446 〈/2ekernel〉
447 〈*2ekernel | \latexrelease〉
448 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
449 〈\latexrelease〉 {\centering}{Set finalhyphendemerits}%

```

\centering

```

450 \DeclareRobustCommand\centering{%
451 \let\\@centercr
452 \rightskip\@flushglue\leftskip\@flushglue
453 \finalhyphendemerits=\z@
454 \parindent\z@\parfillskip\z@skip}

```

(End of definition for \centering.)

```

\raggedright
455 \DeclareRobustCommand\raggedright{%
456 \let\\@centercr\@rights skip\@flushglue \rights skip\@rights skip
457 \finalhyphendemerits=\z@
458 \leftskip\z@skip
459 \parindent\z@}

(End of definition for \raggedright.)

\raggedleft
460 \DeclareRobustCommand\raggedleft{%
461 \let\\@centercr
462 \rights skip\z@skip\leftskip\@flushglue
463 \finalhyphendemerits=\z@
464 \parindent\z@\parfillskip\z@skip}

(End of definition for \raggedleft.)

465 </2ekernel | latexrelease>
466 <latexrelease>\EndIncludeInRelease
467 <latexrelease>\IncludeInRelease{2019/10/01}%
468 <latexrelease> {\centering}{\Make commands robust}%
469 <latexrelease>
470 <latexrelease>\DeclareRobustCommand\centering{%
471 <latexrelease> \let\\@centercr
472 <latexrelease> \rights skip\@flushglue\leftskip\@flushglue
473 <latexrelease> \parindent\z@\parfillskip\z@skip}
474 <latexrelease>\DeclareRobustCommand\raggedright{%
475 <latexrelease> \let\\@centercr\rights skip\@flushglue \rights skip\@rights skip
476 <latexrelease> \leftskip\z@skip
477 <latexrelease> \parindent\z@}
478 <latexrelease>\DeclareRobustCommand\raggedleft{%
479 <latexrelease> \let\\@centercr
480 <latexrelease> \rights skip\z@skip\leftskip\@flushglue
481 <latexrelease> \parindent\z@\parfillskip\z@skip}
482 <latexrelease>\EndIncludeInRelease
483 <latexrelease>
484 <latexrelease>\IncludeInRelease{0000/00/00}%
485 <latexrelease> {\centering}{\Make commands robust}%
486 <latexrelease>
487 <latexrelease>\kernel@make@fragile\centering
488 <latexrelease>\kernel@make@fragile\raggedright
489 <latexrelease>\kernel@make@fragile\raggedleft
490 <latexrelease>
491 <latexrelease>\EndIncludeInRelease
492 <*2ekernel>

{@rights skip
493 \newskip\@rights skip \@rights skip \z@skip

(End of definition for \@rights skip.)
```

**flushleft** (*env.*) We use `\relax` to prevent `\item` scanning too far.

```

494 \def\flushleft{\trivlist \raggedright\item\relax}
495 \def\endflushleft{\endtrivlist}
```

`flushright` (*env.*) We use `\relax` to prevent `\item` scanning too far.

```
496 \def\flushright{\trivlist \raggedleft\item\relax}
497 \def\endflushright{\endtrivlist}
```

### 1.3 Verbatim

```
498 \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
\@vobeytabs
499 </2ekernel>
500 <|latexrelease|\IncludeInRelease{2023/11/01}%
501 <|latexrelease| {\@vobeytabs}{Obeyed tabs}%
502 <*2ekernel | latexrelease>
503 {\catcode`\\ =\active%
504 \gdef\@vobeyspaces{\catcode`\ \\ active\let \@xobeysp\@vobeytabs}%
505 {\catcode`\\^I=\active%
506 \gdef\@vobeytabs{\catcode`\\^I\active\let ^I\@xobeytab}%
507 \global\let ^I=\space
508 }
509 </2ekernel | latexrelease>
510 <|latexrelease|\EndIncludeInRelease
511 <|latexrelease|\IncludeInRelease{0000/00/00}%
512 <|latexrelease| {\@vobeytabs}{Obeyed tabs}%
513 <|latexrelease|{\catcode`\\ =\active%
514 <|latexrelease|\gdef\@vobeyspaces{\catcode`\ \\ active\let \@xobeysp}%
515 <|latexrelease|\let \@vobeytabs\@undefined
516 <|latexrelease|\EndIncludeInRelease
517 <*2ekernel>
```

(*End of definition for `\@vobeyspaces` and `\@vobeytabs`.*)

```
\@xobeysp
```

(*End of definition for `\@xobeysp`.*)

```
\@xverbatim
\@sxverbatim
518 \begingroup \catcode `|=0 \catcode '['=1
519 \catcode']=2 \catcode '\={12 \catcode '`'=12
520 \catcode`\\=12 \gdef\@xverbatim#1\end{verbatim}[#1\end{verbatim}]
521 \gdef\@sxverbatim#1\end{verbatim*}[#1\end{verbatim*}]
522 \endgroup
```

(*End of definition for `\@xverbatim` and `\@sxverbatim`.*)

@verbatim Real start of verbatim environment We use \relax to prevent \item scanning too far.

```
523 </2ekernel>
524 <*2ekernel | latexrelease>
525 <latexrelease>\IncludeInRelease{2017-04-15}{\@verbatim}%
526 <latexrelease> {Disable hyphenation in verbatim}%
527 \def\@verbatim{\trivlist \item\relax
528 \if@minipage\else\vskip\parskip\fi
529 \leftskip\@totalleftmargin\rightskip\z@skip
530 \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.

```
531 \@@par
```

Set \language here to suppress hyphenation. Done this way rather than setting \hyphenchar as that is a global setting.

```
532 \language\l@nohyphenation
533 \@tempswafalse
534 \def\part{%
535 \if@tempswa
```

A \leavevmode added: needed if, for example, a blank verbatim line is the first thing in a list item (wow!).

```
536 \leavevmode \null \@@par\penalty\interlinepenalty
537 \else
538 \@tempswatrue
539 \ifhmode\@@par\penalty\interlinepenalty\fi
540 \fi} %
```

To allow customization we hide the font used in a separate macro.

```
541 \let\do\@makeother \dospecials
542 \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!

```
543 \everypar \expandafter{\the\everypar \unpenalty}%
544 }
545 </2ekernel | latexrelease>
546 <latexrelease>\EndIncludeInRelease
547 <latexrelease>\IncludeInRelease{0000-00-00}{\@verbatim}%
548 <latexrelease> {Disable hyphenation in verbatim}%
549 <latexrelease>\def\@verbatim{\trivlist \item\relax
550 <latexrelease> \if@minipage\else\vskip\parskip\fi
551 <latexrelease> \leftskip\@totalleftmargin\rightskip\z@skip
552 <latexrelease> \parindent\z@\parfillskip\@flushglue\parskip\z@skip
553 <latexrelease> \@@par
554 <latexrelease> \@tempswafalse
555 <latexrelease> \def\par{%
556 \if@tempswa
557 \leavevmode \null \@@par\penalty\interlinepenalty
558 \else
559 \@tempswatrue
560 \ifhmode\@@par\penalty\interlinepenalty\fi
561 \fi} %
562 <latexrelease> \let\do\@makeother \dospecials
```

```

563 <|latexrelease> \obeylines \verbatim@font \onoligs
564 <|latexrelease> \hyphenchar\font\m@ne
565 <|latexrelease> \everypar \expandafter{\the\everypar \unpenalty}%
566 <|latexrelease>
567 <|latexrelease>\EndIncludeInRelease
568 <|2ekernel>

(End of definition for \verbatim.)
```

**\verb+\endverbatim+** (RmS 93/09/19) Protected against ‘missing item’ error message triggered by empty verbatim environment.

```

569 \def\verbatim{\@verbatim \frenchspacing\@vobeyspaces \xverbatim}
570 \def\endverbatim{\if@newlist \leavevmode\fi\endtrivlist}
(End of definition for \verbatim and \endverbatim.)
```

**\verb+@font+** Macro to select the font used for verbatim typesetting. It also does other work if necessary for the font used.

```
571 \def\verb+@font+{\normalfont\ttfamily}
```

(End of definition for \verb+@font+.)

```

572 <|2ekernel>
573 <|2ekernel | latexrelease>
574 <|latexrelease>\IncludeInRelease{2018/12/01}%
575 <|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

```
576 \DeclareRobustCommand\asciispace{\char 32 }
```

(End of definition for \asciispace.)

**\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.

```

577 \ifx\Umathcode\@undefined
578 \let\verbvisiblespace\asciispace % Pdftex version
579 \else
580 \DeclareRobustCommand\verbvisiblespace
581 {\leavevmode{\usefont{OT1}{cmtt}{m}{n}\asciispace}} % xetex/luatex version
582 \fi
```

(End of definition for \verbvisiblespace.)

**\@verbvisiblespacebox** The box to hold the visible space character if it isn’t in slot 32 in the current typewriter font.

```
583 \newbox\@verbvisiblespacebox
```

(End of definition for \@verbvisiblespacebox.)

**\verb+\*+(env)** For \verb+\*+ we also set up the correct visible space character definition and then run \verb+\@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.

```

584 \@namedef{verb+*+}{\@verbatim
585 \@setupverbvisiblespace
586 \frenchspacing\@vobeyspaces\@sxverbatim}
587 \expandafter\let\csname endverb+*+\endcsname =\endverbatim
```

```

588 {/2ekernel | latexrelease}
589 <|latexrelease|\EndIncludeInRelease
590 <|latexrelease|\IncludeInRelease{0000/00/00}%
591 <|latexrelease| {\verbvisible}{Setup visible space for \verb}%
592 <|latexrelease|
593 <|latexrelease|\@namedef{verbatim*}{\@verbatim\@xverbatim}
594 <|latexrelease|
595 <|latexrelease|\let\asciispace \@undefined
596 <|latexrelease|\let\verbvisible \@undefined
597 <|latexrelease|\let\@setupverbvisible\@undefined
598 <|latexrelease|\let\@verbvisiblebox \@undefined
599 <|latexrelease|\EndIncludeInRelease
600 {*2ekernel}

```

\@setupverbvisible In pdf $\text{\TeX}$  a catcode 12 space will produce the character in slot 32 which is assumed to be a visible space character (in a typewriter font in OT1 or T1 encoding). In Xe $\text{\TeX}$  or Lua $\text{\TeX}$  a font in TU encoding is normally used and that has a real space in this slot. So what we do in this case is this: we check the definition of \verbvisible and if it is \asciispace we assume that the char32 can be used (e.g., in pdft $\text{\TeX}$ ). We then redefine \xobeysp so that after running \vobeyspaces we get characters from slot 32 for each active space.

```

601 {/2ekernel}
602 {*2ekernel | latexrelease}
603 <|latexrelease|\IncludeInRelease{2023/11/01}%
604 <|latexrelease| {\@setupverbvisible}{Setup visible tab for \verb}%
605 \def\@setupverbvisible{%
606 \ifx\verbvisible\asciispace
607 \let\xobeysp\asciispace
608 \else

```

Otherwise we measure the width of a character in the mono-spaced current font and place a \verbvisible 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 \verbvisible one could use, for example, the \textvisible of the current typewriter font.

```

609 \setbox\z@\hbox{x}%
610 \setbox\@verbvisiblebox\hbox to\wd\z@{\hss\verbvisible\hss}%
611 \def\xobeysp{\leavevmode\copy\verbvisiblebox}%
612 \fi
613 \@setupverbvisibletab
614 }
615 {/2ekernel | latexrelease}
616 <|latexrelease|\EndIncludeInRelease
617 <|latexrelease|\IncludeInRelease{2018/12/01}%
618 <|latexrelease| {\@setupverbvisible}{Setup visible space for \verb}%
619 <|latexrelease|\def\@setupverbvisible{%
620 <|latexrelease| \ifx\verbvisible\asciispace
621 <|latexrelease| \let\xobeysp\asciispace
622 <|latexrelease| \else
623 <|latexrelease| \setbox\z@\hbox{x}%
624 <|latexrelease| \setbox\@verbvisiblebox\hbox to\wd\z@{\hss\verbvisible\hss}%
625 <|latexrelease| \def\xobeysp{\leavevmode\copy\verbvisiblebox}%
626 <|latexrelease| \fi

```

```

627 \{latexrelease\}
628 \{latexrelease\}\EndIncludeInRelease
629 \{latexrelease\}\IncludeInRelease{0000/00/00}%
630 \{latexrelease\} {\@setupverbvisiblespace}{Setup visible space for \verb}%
631 \{latexrelease\}\let\@setupverbvisiblespace\undefined
632 \{latexrelease\}\EndIncludeInRelease
633 {*2ekernel}

```

(End of definition for \@setupverbvisiblespace.)

\@setupverbvisibletab A redirection: just a simple wrapper.

```

634 \{/2ekernel\}
635 \{latexrelease\}\IncludeInRelease{2023/11/01}%
636 \{latexrelease\} {\@setupverbvisibletab}{Setup visible tab for \verb}%
637 {*2ekernel | latexrelease}
638 \def\@setupverbvisibletab{\let\xobeytab\xobeysp}
639 \{/2ekernel | latexrelease}
640 \{latexrelease\}\EndIncludeInRelease
641 \{latexrelease\}\IncludeInRelease{0000/00/00}%
642 \{latexrelease\} {\@setupverbvisibletab}{Setup visible tab for \verb}%
643 \{latexrelease\}\let\@setupverbvisibletab\undefined
644 \{latexrelease\}\EndIncludeInRelease
645 {*2ekernel}

```

(End of definition for \@setupverbvisibletab.)

\@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.

```

646 \{/2ekernel\}
647 {*2ekernel | latexrelease}
648 \{latexrelease\}\IncludeInRelease{2023/11/01}%
649 \{latexrelease\} {\@sverb}{Support visible tabs}%

```

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).

```

650 \def\@sverb#1{\if\noexpand#1 \expandafter\@sverb\else\@@sverb{#1}\fi}

651 \def\@@sverb#1{%
652 \catcode`#1\active
653 \lccode`\~`#1%
654 \gdef\verb@balance@group{\verb@egroup
655 \@latex@error{\noexpand\verb illegal in argument}\@ehc}%
656 \aftergroup\verb@balance@group
657 \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.

```

658 \ifnum0%
659 \ifnum\catcode`\ =\active\else 1\fi
660 \ifnum\catcode`\^=I=\active\else 1\fi

```

```

661 =0 %
662 \else \@setupverbvisiblespace \vobeyspaces \fi
663 }

664 </2ekernel | latexrelease>
665 <latexrelease>\EndIncludeInRelease
666 <latexrelease>\IncludeInRelease{2020/10/01}%
667 <latexrelease> {\@sverb}{Drop spaces before \verb delimiter}%
668 <latexrelease>\def\@sverb#1{%
669 <latexrelease> \catcode`#1\active
670 <latexrelease> \lccode`\~`#1%
671 <latexrelease> \gdef\verb@balance@group{\verb@egroup
672 <latexrelease> \o@late@error{\noexpand\verb illegal in argument}\@ehc}%
673 <latexrelease> \aftergroup\verb@balance@group
674 <latexrelease> \lowercase{\let~\verb@egroup}%
675 <latexrelease> \ifnum\catcode`\ =\active
676 <latexrelease> \else \@setupverbvisiblespace \vobeyspaces \fi
677 <latexrelease>}

678 <latexrelease>\EndIncludeInRelease
679 <latexrelease>\IncludeInRelease{2018/12/01}%
680 <latexrelease> {\@sverb}{Setup visible space for \verb}%
681 <latexrelease>
682 <latexrelease>\def\@sverb#1{%
683 <latexrelease> \catcode`#1\active
684 <latexrelease> \lccode`\~`#1%
685 <latexrelease> \gdef\verb@balance@group{\verb@egroup
686 <latexrelease> \o@late@error{\noexpand\verb illegal in command argument}\@ehc}%
687 <latexrelease> \aftergroup\verb@balance@group
688 <latexrelease> \lowercase{\let~\verb@egroup}%
689 <latexrelease> \ifnum\catcode`\ =\active
690 <latexrelease> \else \@setupverbvisiblespace \vobeyspaces \fi
691 <latexrelease>}

692 <latexrelease>\let\@sverb\undefined
693 <latexrelease>\EndIncludeInRelease
694 <latexrelease>
695 <latexrelease>\IncludeInRelease{0000/00/00}%
696 <latexrelease> {\@sverb}{Setup visible space for \verb}%
697 <latexrelease>\def\@sverb#1{%
698 <latexrelease> \catcode`#1\active
699 <latexrelease> \lccode`\~`#1%
700 <latexrelease> \gdef\verb@balance@group{\verb@egroup
701 <latexrelease> \o@late@error{\noexpand\verb illegal in command argument}\@ehc}%
702 <latexrelease> \aftergroup\verb@balance@group
703 <latexrelease> \lowercase{\let~\verb@egroup}%
704 <latexrelease>}
705 <latexrelease>\EndIncludeInRelease
706 {*2ekernel}

```

(End of definition for `\@sverb` and `\@sverb`.)

```
\@makeother
707 \def\@makeother#1{\catcode`\#12\relax}
```

(End of definition for `\@makeother`.)

```

\verb@balance@group
 708 \let\verb@balance@group\@empty
 (End of definition for \verb@balance@group.)
```

```

\verb@egroup
 709 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
 (End of definition for \verb@egroup.)
```

```

\verb@eol@error
 710 \begingroup
 711 \obeylines%
 712 \gdef\verb@eol@error{\obeylines%
 713 \def^~M{\verb@egroup\@latex@error{%
 714 \noexpand\verb ended by end of line}\@ehc}}%
 715 \endgroup
 (End of definition for \verb@eol@error.)
```

**\verb** Typesetting a small piece verbatim.

```

 716 </2ekernel>
 717 <*2ekernel | latexrelease>
 718 <latexrelease>\IncludeInRelease{2017-04-15}{\verb}%
 719 <latexrelease> {Disable hyphenation in verb}%
 720 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
 721 \bgroup
 722 \verb@eol@error \let\do\@makeother \dospecials
 723 \verbatim@font\@noligs
```

Set `\language` here to suppress hyphenation. Done this way rather than setting `\hyphenchar` as that is a global setting.

```

 724 \language\l@nohyphenation
 725 \@ifstar\@sverb\@verb}
 726 </2ekernel | latexrelease>
 727 <latexrelease>\EndIncludeInRelease
 728 <latexrelease>\IncludeInRelease{0000-00-00}{\verb}%
 729 <latexrelease> {Disable hyphenation in verb}%
 730 <latexrelease>\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
 731 <latexrelease> \bgroup
 732 <latexrelease> \verb@eol@error \let\do\@makeother \dospecials
 733 <latexrelease> \verbatim@font\@noligs
 734 <latexrelease> \@ifstar\@sverb\@verb}
 735 <latexrelease>\EndIncludeInRelease
 736 <*2ekernel>
```

(End of definition for \verb.)

```

\@verb
 737 \def\@verb{\@vobeyspaces \frenchspacing \@sverb}
 (End of definition for \@verb.)
```

```

\verbatim@nolig@list
 738 \def\verbatim@nolig@list{\do\`{\do\<\do\>\do\,\do\-\}}
```

(End of definition for \verbatim@nolig@list.)

```
\do@noligs
739 \def\do@noligs#1{%
740 \catcode`#1\active
741 \begingroup
742 \lccode`\~`#1\relax
743 \lowercase{\endgroup\def~{\leavevmode\kern\z@\char`#1}}}
```

(End of definition for \do@noligs.)

\@noligs To stay compatible with packages that use \@noligs we keep it.

```
744 \def\@noligs{\let\do\do@noligs \verbatim@nolig@list}
```

(End of definition for \@noligs.)

```
745 </2ekernel>
```

# File 38

## ltmath.dtx

### 1 Math setup

This file contains a lot of the original plain T<sub>E</sub>X code, as well as the L<sup>A</sup>T<sub>E</sub>X environments for math. It still needs sorting out.

```
1 <*2ekernel>
2 \message{math definitions,}
```

#### 1.1 Math commands based on plain T<sub>E</sub>X

##### 1.1.1 The log-like functions

\log The standard operators:

```
3 \DeclareRobustCommand\log{\mathop{\operator@font log}\nolimits}
4 \DeclareRobustCommand\lg{\mathop{\operator@font lg}\nolimits}
5 \DeclareRobustCommand\ln{\mathop{\operator@font ln}\nolimits}
6 \DeclareRobustCommand\lim{\mathop{\operator@font lim}\nolimits}
7 \DeclareRobustCommand\limsup{\mathop{\operator@font lim\,,sup}\nolimits}
8 \DeclareRobustCommand\liminf{\mathop{\operator@font lim\,,inf}\nolimits}
9 \DeclareRobustCommand\sin{\mathop{\operator@font sin}\nolimits}
10 \DeclareRobustCommand\arcsin{\mathop{\operator@font arcsin}\nolimits}
11 \DeclareRobustCommand\sinh{\mathop{\operator@font sinh}\nolimits}
12 \DeclareRobustCommand\cos{\mathop{\operator@font cos}\nolimits}
13 \DeclareRobustCommand\arccos{\mathop{\operator@font arccos}\nolimits}
14 \DeclareRobustCommand\cosh{\mathop{\operator@font cosh}\nolimits}
15 \DeclareRobustCommand\tan{\mathop{\operator@font tan}\nolimits}
16 \DeclareRobustCommand\arctan{\mathop{\operator@font arctan}\nolimits}
17 \DeclareRobustCommand\tanh{\mathop{\operator@font tanh}\nolimits}
18 \DeclareRobustCommand\cot{\mathop{\operator@font cot}\nolimits}
19 \DeclareRobustCommand\coth{\mathop{\operator@font coth}\nolimits}
20 \DeclareRobustCommand\sec{\mathop{\operator@font sec}\nolimits}
21 \DeclareRobustCommand\csc{\mathop{\operator@font csc}\nolimits}
22 \DeclareRobustCommand\max{\mathop{\operator@font max}\nolimits}
23 \DeclareRobustCommand\min{\mathop{\operator@font min}\nolimits}
24 \DeclareRobustCommand\sup{\mathop{\operator@font sup}\nolimits}
25 \DeclareRobustCommand\inf{\mathop{\operator@font inf}\nolimits}
26 \DeclareRobustCommand\arg{\mathop{\operator@font arg}\nolimits}
27 \DeclareRobustCommand\ker{\mathop{\operator@font ker}\nolimits}
28 \DeclareRobustCommand\dim{\mathop{\operator@font dim}\nolimits}
29 \DeclareRobustCommand\hom{\mathop{\operator@font hom}\nolimits}
30 \DeclareRobustCommand\det{\mathop{\operator@font det}\nolimits}
31 \DeclareRobustCommand\exp{\mathop{\operator@font exp}\nolimits}
32 \DeclareRobustCommand\Pr{\mathop{\operator@font Pr}\nolimits}
33 \DeclareRobustCommand\gcd{\mathop{\operator@font gcd}\nolimits}
34 \DeclareRobustCommand\deg{\mathop{\operator@font deg}\nolimits}
```

(End of definition for \log.)

\bmod And some operators have to be done by hand:

```

35 \DeclareRobustCommand\bmod{%
36 \nonscript\mskip-\medmuskip\mkern5mu%
37 \mathbin{\operator@font mod}\penalty900\mkern5mu%
38 \nonscript\mskip-\medmuskip}
(End of definition for \bmod)

\pmod
39 \DeclareRobustCommand\pmod[1]{%
40 \allowbreak\mkern18mu(\operator@font mod)\,,\,#1)}
(End of definition for \pmod.)

```

### 1.1.2 Biggggg

\big Variants on \big and friends for use with delimiters:

```

41 \DeclareRobustCommand\bigl{\mathopen\big}
42 \DeclareRobustCommand\bigm{\mathrel\big}
43 \DeclareRobustCommand\bigr{\mathclose\big}
44 \DeclareRobustCommand\Bigl{\mathopen\Big}
45 \DeclareRobustCommand\Bigm{\mathrel\Big}
46 \DeclareRobustCommand\Bigr{\mathclose\Big}
47 \DeclareRobustCommand\biggl{\mathopen\bigg}
48 \DeclareRobustCommand\biggm{\mathrel\bigg}
49 \DeclareRobustCommand\biggr{\mathclose\bigg}
50 \DeclareRobustCommand\Biggl{\mathopen\Bigg}
51 \DeclareRobustCommand\Biggm{\mathrel\Bigg}
52 \DeclareRobustCommand\Biggr{\mathclose\Bigg}

```

(End of definition for \big.)

### 1.1.3 The UNSORTED Rest

The other math commands are lifted from plain TeX.

\jot

```

53 \newdimen\jot
54 \jot=3pt

```

(End of definition for \jot.)

\interdisplaylinepenalty

```

55 \newcount\interdisplaylinepenalty
56 \interdisplaylinepenalty=100

```

(End of definition for \interdisplaylinepenalty.)

\choose

```

57 \def\choose{\atopwithdelims()}

```

(End of definition for \choose.)

\brack

```

58 \def\brack{\atopwithdelims[]}

```

(End of definition for \brack.)

```

\brace
59 \def\brace{\atopwithdelims\{\}}
(End of definition for \brace.)

\mathpalette
60 \def\mathpalette#1#2{%
61 \mathchoice
62 {#1\displaystyle{#2}}%
63 {#1\textstyle{#2}}%
64 {#1\scriptstyle{#2}}%
65 {#1\scriptscriptstyle{#2}}}
(End of definition for \mathpalette.)

\root
\rootbox
66 \newbox\rootbox
\root@t
67 \def\root#1{\of{%
68 \setbox\rootbox\hbox{$\m@th\scriptscriptstyle{#1}$}%
69 \mathpalette\root@t}}
\def\root@t#1#2{%
70 \setbox\z@\hbox{$\m@th\sqrt{\#2}$}%
71 \dimen@\ht\z@\advance\dimen@-\dp\z@
72 \mkern5mu\raise.6\dimen@\copy\rootbox
73 \mkern-10mu\box\z@}
(End of definition for \root, \rootbox, and \root@t.)

\phantom
\hphantom
75 \newif\ifv@
\phantom
76 \newif\ifh@
77 </2ekernel>
78 <*2ekernel | latexrelease>
79 <latexrelease>\IncludeInRelease{2019/10/01}%
80 <latexrelease> {\vphantom}{Make commands robust}%
81 \DeclareRobustCommand\vphantom{\v@true\h@false\ph@nt}
82 \DeclareRobustCommand\hphantom{\v@false\h@true\ph@nt}
83 \DeclareRobustCommand

84 \DeclareRobustCommand\mathstrut{\vphantom{}}

\mathstrut
85 </2ekernel | latexrelease>
86 <latexrelease>\EndIncludeInRelease
87 <latexrelease>\IncludeInRelease{0000/00/00}%
88 <latexrelease> {\vphantom}{Make commands robust}%
89 <latexrelease>
90 <latexrelease>\kernel@make@fragile\vphantom
91 <latexrelease>\kernel@make@fragile\hphantom
92 <latexrelease>\kernel@make@fragile\phantom
93 <latexrelease>\kernel@make@fragile\mathstrut
94 <latexrelease>
95 <latexrelease>\EndIncludeInRelease
96 <*2ekernel>

```

```

97 \def\ph@nt{%
98 \ifmmode
99 \expandafter\mathpalette\expandafter\mathph@nt
100 \else
101 \expandafter\makeph@nt
102 \fi}
103 \def\makeph@nt#1{%
104 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finph@nt}
105 \def\mathph@nt#1#2{%
106 \setbox\z@\hbox{$\m@th#1{#2}$}\finph@nt}
107 </2ekernel>
108 <*2ekernel | latexrelease>
109 <latexrelease>\IncludeInRelease{2018/12/01}%
110 <latexrelease> {\finph@nt}{Start LR-mode}%
111 \def\finph@nt{%
112 \setbox\tw@\null
113 \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
114 \ifh@ \wd\tw@\wd\z@\fi
115 \leavevmode@ifvmode\box\tw@}
116 </2ekernel | latexrelease>
117 <latexrelease>\EndIncludeInRelease
118 <latexrelease>\IncludeInRelease{0000/00/00}%
119 <latexrelease> {\finph@nt}{Start LR-mode}%
120 <latexrelease>\def\finph@nt{%
121 \setbox\tw@\null
122 \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
123 \ifh@ \wd\tw@\wd\z@\fi \box\tw@}
124 <latexrelease>\EndIncludeInRelease
125 <*2ekernel>

```

(End of definition for `\phantom` and others.)

```

\smash
126 \DeclareRobustCommand\smash{%
127 \relax % \relax, in case this comes first in \halign
128 \ifmmode
129 \expandafter\mathpalette\expandafter\mathsm@sh
130 \else
131 \expandafter\makesm@sh
132 \fi}
133 \def\makesm@sh#1{%
134 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
135 </2ekernel>
136 <*2ekernel | latexrelease>
137 <latexrelease>\IncludeInRelease{2022/11/01}%
138 <latexrelease> {\mathsm@sh}{Guard against reboxing}%
139 \def\mathsm@sh#1#2{%
140 \setbox\z@\hbox{$\m@th#1{#2}$}%

```

The empty brace groups in front of the smashed box (which is placed by `\finsm@sh`) ensures that a `\smash` in math is not just producing a single box with its dimensions altered, but a box plus this second ord atom. The reason is that TeX sometimes reboxes

a box if its the only thing in a place like the denominator of a fraction. This would then undo the smashing and the additional ord atom prevents that. Two ord atoms in a row do not alter the horizontal spacing in a formula so this is otherwise transparent.

```

141 {}\\finsm@sh}
142 </2ekernel | latexrelease>
143 <latexrelease>\\EndIncludeInRelease
144 <latexrelease>\\IncludeInRelease{0000/00/00}%
145 <latexrelease> {\\mathsm@sh}{Guard against reboxing}%
146 <latexrelease>\\def\\mathsm@sh#1#2{%
147 <latexrelease> \\setbox\\z@\\hbox{\\m@th#1{#2}}\\finsm@sh}
148 <latexrelease>\\EndIncludeInRelease
149 {*2ekernel}

150 </2ekernel>
151 {*2ekernel | latexrelease}
152 <latexrelease>\\IncludeInRelease{2018/12/01}%
153 <latexrelease> {\\finsm@sh}{Start LR-mode}%
154 \\def\\finsm@sh{\\ht\\z@\\z@ \\dp\\z@\\z@ \\leavevmode@ifvmode\\box\\z@}
155 </2ekernel | latexrelease>
156 <latexrelease>\\EndIncludeInRelease
157 <latexrelease>\\IncludeInRelease{0000/00/00}%
158 <latexrelease> {\\finsm@sh}{Start LR-mode}%
159 <latexrelease>\\def\\finsm@sh{\\ht\\z@\\z@ \\dp\\z@\\z@ \\box\\z@}
160 <latexrelease>\\EndIncludeInRelease
161 {*2ekernel}

```

(*End of definition for \smash.*)

### \buildrel

```

162 \\def\\buildrel#1\\over#2{\\mathrel{\\mathop{\\kern\\z@#2}\\limits^{\\#1}}}

```

(*End of definition for \buildrel.*)

```

163 </2ekernel>
164 {*2ekernel | latexrelease}
165 <latexrelease>\\IncludeInRelease{2019/10/01}%
166 <latexrelease> {\\cases}{Make commands robust}%

```

### \cases

```

167 \\DeclareRobustCommand*\\cases[1]{\\left\\{\\,\\,\\vcenter{\\normalbaselines\\m@th
168 \\ialign{$##\\hfil$&\\quad{##}\\hfil\\crcr#1\\crcr}}\\right.}

```

(*End of definition for \cases.*)

### \matrix

```

169 \\DeclareRobustCommand*\\matrix[1]{\\null\\,\\vcenter{\\normalbaselines\\m@th
170 \\ialign{\\hfil##\\hfil&&\\quad\\hfil##\\hfil\\crcr
171 \\mathstrut\\crcr\\noalign{\\kern-\\baselineskip}
172 #1\\crcr\\mathstrut\\crcr\\noalign{\\kern-\\baselineskip}}}\\,}

```

(*End of definition for \matrix.*)

### \pmatrix

```

173 \\DeclareRobustCommand*\\pmatrix[1]{\\left(\\matrix{\\right)}

```

```

(End of definition for \pmatrix.)

174 </2ekernel | latexrelease>
175 <latexrelease>\EndIncludeInRelease
176 <latexrelease>\IncludeInRelease{0000/00/00}%
177 <latexrelease> {\cases}{\Make commands robust}%
178 <latexrelease>
179 <latexrelease>\kernel@make@fragile\cases
180 <latexrelease>\kernel@make@fragile\matrix
181 <latexrelease>\kernel@make@fragile\pmatrix
182 <latexrelease>
183 <latexrelease>\EndIncludeInRelease
184 {*2ekernel}

\bordermatrix
185 \def\bordermatrix#1{\begingroup \m@th
186 \tempdima 8.75\p@
187 \setbox\z@\vbox{%
188 \def\cr{\crcr\noalign{\kern2\p@\global\let\cr\endline}}%
189 \ialign{$##$\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
190 \quad\hfil$##$\hfil\crcr
191 \omit\strut\hfil\crcr\noalign{\kern-\baselineskip}%
192 #1\crcr\omit\strut\cr}%
193 \setbox\tw@\vbox{\unvcopy\z@\global\setbox\one\lastbox}%
194 \setbox\tw@\hbox{\unhbox\one\unskip\global\setbox\one\lastbox}%
195 \setbox\tw@\hbox{$\kern\wd\one\kern-\@tempdima\left(\kern-\wd\one
196 \global\setbox\one\vbox{\box\one\kern2\p@}%
197 \vcenter{\kern-\ht\one\unvbox\z@\kern-\baselineskip}\,,\right)$}%
198 \null;\vbox{\kern\ht\one\box\tw@}\endgroup

(End of definition for \bordermatrix.)

\openup
199 \protected\def\openup{\afterassignment\openup\dimen@}
200 \def\openup{\advance\lineskip\dimen@
201 \advance\baselineskip\dimen@
202 \advance\lineskiplimit\dimen@}

(End of definition for \openup.)

\displaylines
203 \newif\ifdt@p
204 \def\displ@y{\global\dt@ptrue\openup\jot\m@th
205 \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
206 \vskip-\lineskiplimit \vskip\normalineskiplimit \fi
207 \else \penalty\interdisplaylinepenalty \fi}}}
208 \def\@lign{\tabskip\z@skip\everycr{}% restore inside \displ@y
209 \def\displaylines#1{\displ@y \tabskip\z@skip
210 \halign{\hb@xt@{\displaywidth{$\@lign\hfil\displaystyle##\hfil$}}\crcr
211 #1\crcr}}
```

(End of definition for \displaylines.)

```

\sp
\sb 212 \let\sp=^
 213 \let\sb=_
```

(End of definition for `\sp` and `\sb`.)

```

\tmspace
\thinspace
\!_
\negthinspace
\:
\medspace
\negmedspace
\;
\thickspace
\negthickspace
```

Originally L<sup>A</sup>T<sub>E</sub>X only provided a small set of spacing commands for use in text and math, some of the commands like `\;` were only supported in math 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.

```

214 </2ekernel>
215 <*2ekernel | latexrelease>
216 <latexrelease>\IncludeInRelease{2020/10/01}%
217 <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.
```

```

218 \DeclareRobustCommand\tmspace[3]{%
219 \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.
220 \DeclareRobustCommand\,{\tmspace+\thinmuskip{.16667em}}
221 \let\thinspace\,
222 \DeclareRobustCommand\!{\tmspace-\thinmuskip{.16667em}}
223 \let\negthinspace\!
224 \DeclareRobustCommand\:{\tmspace+\medmuskip{.2222em}}
225 \let\medspace\:
```

L<sup>A</sup>T<sub>E</sub>X has a second name for this in its manual:

```

226 \let\>=\:
227 \DeclareRobustCommand\negmedspace{\tmspace-\medmuskip{.2222em}}
228 \DeclareRobustCommand\,{\tmspace+\thickmuskip{.2777em}}
229 \let\thickspace\;
230 \DeclareRobustCommand\negthickspace{\tmspace-\thickmuskip{.2777em}}
231 </2ekernel | latexrelease>
232 <latexrelease>\EndIncludeInRelease

233 <latexrelease>\IncludeInRelease{0000/00/00}%
234 <latexrelease> {\tmspace}{amsmath spacing commands}%
235 <latexrelease>
236 <latexrelease>\let\tmspace\@undefined
237 <latexrelease>\DeclareRobustCommand\,{\,}{%
238 <latexrelease> \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi}
239 <latexrelease>\DeclareRobustCommand\thinspace{\leavevmode@ifvmode\kern .16667em }
240 <latexrelease>\DeclareRobustCommand\negthinspace{\leavevmode@ifvmode\kern-.16667em }
241 <latexrelease>\def\>{\mskip\medmuskip}
242 <latexrelease>\let\>=\>
243 <latexrelease>\def\,{\mskip\thickmuskip}
244 <latexrelease>\def\!{\mskip-\thinmuskip}
245 <latexrelease>
```

```

246 ⟨latexrelease⟩\let\negmedspace\@undefined
247 ⟨latexrelease⟩\let\negthickspace\@undefined
248 ⟨latexrelease⟩
249 ⟨latexrelease⟩\EndIncludeInRelease
250 {*2ekernel}

(End of definition for \tmspace and others.)

*
251 \DeclareRobustCommand*{\discretionary{\thinspace}{\the\textfont2\char2}{}{}}

(End of definition for *.)

\: Nickname for the medium space since \> is not available inside tabbing.
252 \%let\:=\>

(End of definition for \::)

\active@math@prime This is the definition of the active math prime.
253 \def\active@math@prime{^\bgroup\prim@s}

(End of definition for \active@math@prime.)

\prime@s
254 {\catcode`'=active \global\let'\active@math@prime}
255 \def\prim@s{%
256 \prime\futurelet\let@token\pr@m@s}
257 \def\pr@m@s{%
258 \ifx'\let@token
259 \expandafter\pr@@s
260 \else
261 \ifx`\let@token
262 \expandafter\expandafter\expandafter\pr@@t
263 \else
264 \egroup
265 \fi
266 \fi}
267 \def\pr@@s{\prim@s}
268 \def\pr@@t{\#2\egroup}

(End of definition for \prime@s.)

269 {\catcode`_=active \gdef_`_} % _ in math is
270 % 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 \(\.

```

271 </2ekernel>
272 <| latexrelease>\IncludeInRelease{2015/01/01}{\{}{\Make \(\ robust}\%
273 {*2ekernel | latexrelease}
274 \DeclareRobustCommand\(\{%
275 \relax\ifmmode\@badmath\else$\fi\}%
276 \DeclareRobustCommand\){%
277 \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi\}%
278 </2ekernel | latexrelease>
279 <| latexrelease>\EndIncludeInRelease
280 <| latexrelease>\IncludeInRelease{0000/00/00}{\{}{\Make \(\ robust}\%
281 <| latexrelease>\def\(\{%
282 <| latexrelease> \relax\ifmmode\@badmath\else$\fi\}%
283 <| latexrelease>\expandafter\let\csname\string(\endcsname\@undefined
284 <| latexrelease>\def\){%
285 <| latexrelease> \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi\}%
286 <| latexrelease>\expandafter\let\csname\string) \endcsname\@undefined
287 <| latexrelease>\EndIncludeInRelease
288 {*2ekernel}

```

(End of definition for \(\ and \).)

- \[ 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 \$\$).

```

289 </2ekernel>
290 <| latexrelease>\IncludeInRelease{2015/01/01}{\[]}{\{}{\Make \[robust}\%
291 {*2ekernel | latexrelease}
292 \DeclareRobustCommand\[{\{%
293 \relax\ifmmode
294 \@badmath
295 \else
296 \ifvmode
297 \nointerlineskip
298 \makebox[.6\linewidth]{\}%
299 \fi
300 $$\%$$ BRACE MATCH HACK
301 \fi
302 }\%
303 \DeclareRobustCommand\]{%
304 \relax\ifmmode
305 \ifinner
306 \@badmath
307 \else
308 $$\%$$ BRACE MATCH HACK
309 \fi
310 \else
311 \@badmath
312 \fi
313 \ignorespaces
314 }\%

```

```

315 </2ekernel | latexrelease>
316 <latexrelease>\EndIncludeInRelease
317 <latexrelease>\IncludeInRelease{0000/00/00}{\[]}{\Make \[robust}%
318 <latexrelease>\def\[%
319 <latexrelease> \relax\ifmmode
320 <latexrelease> \C@badmath
321 <latexrelease> \else
322 <latexrelease> \ifvmode
323 <latexrelease> \nointerlineskip
324 <latexrelease> \makebox[.6\linewidth]{}}%
325 <latexrelease> \fi
326 <latexrelease> $$$%$$$ BRACE MATCH HACK
327 <latexrelease> \fi
328 <latexrelease>}%
329 <latexrelease>\expandafter\let\csname\string[\endcsname\@undefined
330 <latexrelease>\def\[]{%
331 <latexrelease> \relax\ifmmode
332 <latexrelease> \ifinner
333 <latexrelease> \C@badmath
334 <latexrelease> \else
335 <latexrelease> $$$%$$$ BRACE MATCH HACK
336 <latexrelease> \fi
337 <latexrelease> \else
338 <latexrelease> \C@badmath
339 <latexrelease> \fi
340 <latexrelease> \ignorespaces
341 <latexrelease>}%
342 <latexrelease>\expandafter\let\csname\string] \endcsname\@undefined
343 <latexrelease>\EndIncludeInRelease
344 <*2ekernel>

```

(End of definition for \[ and \].)

**math (env.)** Disguises for \(...\mathord{)} and \(...\mathord{].}

```

displaymath (env.) 345 \let\math=\(
346 \let\endmath=\)
347 \def\displaymath{\[}
348 \def\enddisplaymath{\]} \C@ignoretrue

```

**equation (env.)** Numbered equations, using the counter \c@equation. Note: The document style must define \theequation etc., and do the appropriate \C@addtoreset. It should also redefine \C@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 \C@eqnnum.)

```

349 \C@definecounter{equation}
350 \def\equation{$$\refstepcounter{equation}}
351 \def\endequation{\eqno \hbox{\C@eqnnum}$$\C@ignoretrue}

```

(End of definition for \c@equation.)

**\C@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.

```

352 \def\C@eqnnum{{\normalfont \normalcolor (\theequation)}}

```

(End of definition for \eqnnum.)

\stackrel A disguise for plain T<sub>E</sub>X's buildrel.

353 \DeclareRobustCommand\stackrel[2]{\mathrel{\mathop{\#2}\limits^{\#1}}}

(End of definition for \stackrel.)

\frac A disguise for plain T<sub>E</sub>X's \over.

354 \DeclareRobustCommand\frac[2]{{\begingroup\over\endgroup\over#2}}

(End of definition for \frac.)

\sqrt Add an optional argument to plain's \sqrt to give the *n*th root of an expression  $\sqrt[n]{e}$ .

\@sqrt \DeclareRobustCommand\sqrt{\ifnextchar[\@sqrt\sqrtsign}

356 \def\@sqrt[#1]{\root #1\of{}}

(End of definition for \sqrt and \@sqrt.)

eqnarray@cnt) Here's the eqnarray environment: Default is for left-hand side of equations to be flushright. To make them flushleft, \let\@eqnse = \hfil.

\@eqpen \newcount\@eqcnt

\@eqnse \newcount\@eqpen

359 \newif\ifeqns\@eqnswtrue

360 \newskip\@centering

361 \@centering = Opt 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.

362 \def\eqnarray{%

363 \stepcounter{equation}%

364 \def\@currentlabel{\p@equation\theequation}%

365 \def\@currentcounter{equation}%

366 \global\@eqnswtrue

367 \m@th

368 \global\@eqcnt\z@

369 \tabskip\@centering

370 \let\\@\eqnccr

371 \$\$\everycr{} \halign to\displaywidth\bgroup

372 \hskip\@centering\\$ \displaystyle\tabskip\z@skip{\#}\\$ \@eqnse

373 \&\global\@eqcnt\@ne\hskip \tw@arraycolsep \hfil{\#}\\$ \hfil

374 \&\global\@eqcnt\tw@ \hskip \tw@arraycolsep

375 \\$ \displaystyle{\#}\\$ \hfil\tabskip\@centering

376 \&\global\@eqcnt\thr@@ \hb@xt@{\z@}\bgroup\hss##\egroup

377 \tabskip\z@skip

378 \cr

379 }

380 \def\endeqnarray{%

381 \@eqnccr

382 \egroup

383 \global\advance\c@equation\m@ne

384 \$\$\@ignoretrue

385 }

386 \let\@eqnse=\relax

(End of definition for `\@eqcnt` and others.)

`\nonumber` Switches off equation numbering.

387 `\def\nonumber{\global\@eqnswfalse}`

(End of definition for `\nonumber`.)

`\@eqncr`

```
388 \protected\def\@eqncr{%
389 {\ifnum0='}\fi
390 \@ifstar{%
391 \global\@eqpen\OM\@yeqncr
392 }{%
393 \global\@eqpen\interdisplaylinepenalty \@yeqncr
394 }%
395 }
396 \def\@yeqncr{\@testopt\@eqncr\z@skip}

397 </2ekernel>
398 <*2ekernel | latexrelease>
399 <latexrelease>\IncludeInRelease{2020/10/01}%
400 <latexrelease> {\@\@eqncr}{eqnarray support calc syntax}%
401 \def\@xeqncr[#1]{%
402 \ifnum0='{\fi}%
403 \@\@eqncr
404 \noalign{\penalty\@eqpen\vskip\jot\@vspace\@calcify{#1}}%
405 }
406 </2ekernel | latexrelease>
407 <latexrelease>\EndIncludeInRelease
408 <latexrelease>\IncludeInRelease{0000/00/00}%
409 <latexrelease> {\@\@eqncr}{eqnarray support calc syntax}%
410 <latexrelease>
411 <latexrelease>\def\@xeqncr[#1]{%
412 \ifnum0='{\fi}%
413 \@\@eqncr
414 \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
415 }
416 <latexrelease>\EndIncludeInRelease
417 <*2ekernel>
```

(End of definition for `\@eqncr`, `\@xeqncr`, and `\@yeqncr`.)

`\@@eqncr`

```
418 \def\@@eqncr{\let\reserved@a\relax
419 \ifcase\@eqcnt \def\reserved@a{& &}\or \def\reserved@a{& }%
420 \or \def\reserved@a{&}\else
421 \let\reserved@a\empty
422 \@latex@error{Too many columns in eqnarray environment}\@ehc\fi
423 \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
424 \global\@eqnswtrue\global\@eqcnt\z@\cr}
```

(End of definition for `\@@eqncr`.)

`\eqnarray*\@seqncr`) Here's the eqnarray\* environment:

```
425 \let\@seqncr=\@eqncr
426 \namedef{eqnarray*}{\protected\def\@eqncr{\nonumber\@seqncr}\eqnarray}
427 \namedef{endeqnarray*}{\nonumber\endeqnarray}
```

(End of definition for \@seqncr.)

`\lefteqn` `\lefteqn{FORMULA}` typesets FORMULA in display math style flushleft in a box of width zero.

```
428 \def\lefteqn#1{\rlap{$\displaystyle #1$}}
(End of definition for \lefteqn.)
```

`\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.

```
429 \DeclareRobustCommand{\ensuremath}{%
430 \ifmmode
431 \expandafter\@firstofone
432 \else
433 \expandafter\@ensuredmath
434 \fi}
```

(End of definition for \ensuremath.)

`\@ensuredmath` The `\relax` stops `\ensuremath{}` starting display math.

```
435 \long\def\@ensuredmath#1{$\relax#1$}
```

(End of definition for \@ensuredmath.)

LuaTeX contains new math primitives to place expression over or under horizontally extensible glyphs. Before LuaTeX 1.14 these did not work correctly with the `\mathstyle` primitive and sometimes did not use cramped style in consistent ways. For newer version, we opt into the corrected behavior.

```
436 \ifx\mathdefaultsmode\@undefined\else
437 \mathdefaultsmode=1
438 \fi
```

`\eqno` Ensure the (deprecated) `$$.. \eqno 1 $$` ignores spaces.

```
439 </2ekernel>
440 <*2ekernel | latexrelease>
441 <latexrelease>\IncludeInRelease{2023/06/01}%
442 <latexrelease> {\eqno}{add ignorespaces to eqno}%
443 \let\@kernel@\eqno\eqno
444 \let\@kernel@\leqno\leqno
445 \protected\def\eqno{\@kernel@\eqno\aftergroup\ignorespaces}
446 \protected\def\leqno{\@kernel@\leqno\aftergroup\ignorespaces}
447 </2ekernel | latexrelease>
448 <latexrelease>\EndIncludeInRelease
449 <latexrelease>\IncludeInRelease{0000/00/00}%
450 <latexrelease> {\eqno}{add ignorespaces to eqno}%
451 <latexrelease>\let\eqno\@kernel@\eqno
452 <latexrelease>\let\leqno\@kernel@\leqno
453 <latexrelease>\EndIncludeInRelease
```

(End of definition for `\eqno` and `\leqno`.)

### 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 equation is too wide, the equation number than may overprint the equation itself.

```
454 {*\leqno}
455 \renewcommand{\eqnnum}{\hb@xt@.01\p@{}%
456 \rlap{\normalfont\normalcolor
457 \hskip -\displaywidth(\theequation)}}
458 {/\leqno}
```

(End of definition for `\@eqnnum`.)

#### 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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> 's displayed math environments.

- `\mathindent` The amount of indentation of the equations is stored in a register.

```
459 {*\fleqn}
460 \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.

```
461 \AtEndOfClass{\mathindent\leftmargini}
```

(End of definition for `\mathindent`.)

`\[` Begin display math;

```
462 \IncludeInRelease{2015/01/01}{}{\Make[] robust}%
463 \DeclareRobustCommand[\relax
464 \ifmmode\badmath
465 \else
466 \begin{trivlist}%
467 \begin{parpenalty}\predisplaypenalty
468 \end{parpenalty}\postdisplaypenalty
469 \item[]\leavevmode
470 \hb@xt@\linewidth\bgroup \$\m@th\displaystyle %$
471 \hskip\mathindent\bgroup
472 \fi
473 \EndIncludeInRelease
474 \IncludeInRelease{0000/00/00}{}{\Make[] robust}%
475 \renewcommand[\relax
476 \ifmmode\badmath
477 \else
478 \begin{trivlist}%
479 \begin{parpenalty}\predisplaypenalty
480 \end{parpenalty}\postdisplaypenalty
481 \item[]\leavevmode
```

```

482 \hb@xt@{\linewidth}{\bgroup $ \m@th \displaystyle %$}
483 \hskip\mathindent\bgroup
484 \fi}
485 \EndIncludeInRelease

```

(End of definition for \L.)

\] end display math;

```

486 \IncludeInRelease{2015/01/01}{\L}{\Make \L robust}%
487 \DeclareRobustCommand{\relax
488 \ifmmode
489 \egroup $\hfil% $
490 \egroup
491 \end{trivlist}%
492 \else \badmath
493 \fi}
494 \EndIncludeInRelease
495 \IncludeInRelease{0000/00/00}{\J}{\Make \J robust}%
496 \renewcommand{\relax
497 \ifmmode
498 \egroup $\hfil% $
499 \egroup
500 \end{trivlist}%
501 \else \badmath
502 \fi}
503 \EndIncludeInRelease

```

(End of definition for \J.)

**equation (env.)** The equation environment

```

504 \renewenvironment{equation}%
505 {\begin{parpenalty}\predisplaypenalty
506 \end{parpenalty}\postdisplaypenalty
507 \refstepcounter{equation}%
508 \trivlist \item[] \leavevmode
509 \hb@xt@\linewidth{\bgroup $ \m@th \%$}
510 \displaystyle
511 \hskip\mathindent}%

```

Ensure that there is at least a space between formula and equation number so that they don't bump in each other.

```

512 {$. \hskip .3em minus .3em \hfil \% $%
513 \displaywidth\linewidth\hbox{\eqnnum}%
514 \egroup
515 \endtrivlist}

```

**eqnarray (env.)** The eqnarray environment

```

516 \renewenvironment{eqnarray}%
517 \stepcounter{equation}%
518 \def\currentlabel{\p@equation\theequation}%
519 \def\currentcounter{equation}%
520 \global\eqnswtrue\m@th
521 \global\eqcnt\z@
522 \tabskip\mathindent

```

```
523 \let\\=\\eqnrcr
524 \setlength\abovedisplayskip{\topsep}%
525 \ifvmode
526 \addtolength\abovedisplayskip{\partopsep}%
527 \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.

# File 39

## 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 flexibility in designing the format, including the ability to let the indentation of the first paragraph be different from that of the subsequent ones.

The trivlist environment is equivalent to a list environment whose second argument sets the following parameter values:

`\leftmargin = 0`: causes no indentation of left margin

`\labelwidth = 0`: see below for precise effect this has.

`\itemindent = 0`: with a null label, makes first paragraph have no indentation. Succeeding paragraphs have `\parindent` indentation. To give first paragraph same indentation, set `\itemindent = \parindent` before the `\item[]`.

Every `\item` in a trivlist environment must have an argument—in many cases, this will be the null argument (`\item[]`). The trivlist environment is mainly used for paragraphing environments, like verbatim, in which there is no margin change. It provides the same vertical spacing as the list environment, and works reasonably well when it occurs immediately after an `\item` command in an enclosing list.

## 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.

`\@nmbrrlist` Set true by `\usecounter` command, causes list to be numbered.

`\@listctr` \def'ed by `\usecounter` to name of counter.

`\@noskipsec` A switch set true by a sectioning command when it is creating an in-text heading with `\everypar`.

Throughout a list environment, `\hsize` is the width of the current line, measured from the outermost left margin to the outermost right margin. Environments like tabbing should use `\linewidth` instead of `\hsize`.

Here are the parameters of a list that can be set by commands in the `\list`'s COMMANDS argument. These parameters are all TeX skips or dimensions (defined by `\newskip` or `\newdimen`), so the usual TeX or L<sup>A</sup>T<sub>E</sub>X commands can be used to set them. The commands will be executed in vmode if and only if the `\list` was preceded by a `\par` (or something like an `\end{list}`), so the spacing parameters can be set according to whether the list is inside a paragraph or is its own paragraph.

## 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

\begin{parpenalty}: 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  $\leq$  \labelwidth, then the label is flushed right inside a box of width \labelwidth (with an \hfil). Otherwise, a box of the natural width is employed, which causes an indentation of the text on that line.

\labelsep: space between end of label box and text of first item.

## 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 L<sup>A</sup>T<sub>E</sub>X 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
 \ctrivlist
 \labelwidth := 0
 \leftmargin := 0
 \itemindent := \parindent
 \citemlabel :=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[@citemlabel]
END

\item[LAB] ==

```

```

BEGIN
if @noperitem = true
then @noperitem := false
 % NOTE: then clause hardly every taken,
 % so made a macro \donoperitem
\box@\labels :=G \hbox{\hskip -\leftmargin
\box@\labels
\hskip \leftmargin }

if @minipage = false then
 \tempskipa := \lastskip
 \vskip -\lastskip
 \vskip \tempskipa + \outerparskip - \parskip
fi
else if @inlabel = true
 then \indent \par % previous item empty.
fi
if hmode then 2 \unskip's
 % To remove any space at end of prev.
 % paragraph that could cause a blank line.
\par
fi
if @newlist = T
 then if @nobreak = T % Kludge if list follows \section
 then \addvspace{\outerparskip - \parskip}
 else \addpenalty{\beginparpenalty}
 \addvspace{\topsep}
 \addvspace{\parskip} %% added 4 Sep 85
 fi
 else \addpenalty{\itempenalty}
 \addvspace{\itemsep}
 fi
 @inlabel :=G true
fi

\everypar{ @minipage :=G F
 @newlist :=G F
 if @inlabel = true
 then @inlabel :=G false
 \hskip -\parindent
 \box@\labels
 \penalty 0
 %% 3 Oct 85 - allow line break here
 \box@\labels :=G null
 fi
 \everypar{} }
@nobreak :=G false
if @noitemarg = true
then @noitemarg := false
if @nmbrlist
then \refstepcounter{\listctr}

```

```

 fi fi
 \@tempboxa :=L \hbox{\makelabel{LAB}}
 \box\@labels :=G \@labels \hskip \itemindent
 \hskip - (\labelwidth + \labelsep)
 if \wd \@tempboxa > \labelwidth
 then \box\@tempboxa
 else \hbox to \labelwidth {\unhbox\@tempboxa}
 fi
 \hskip\labelsep
 \ignorespaces %% gobble space up to text
END

```

\makelabel{LABEL} == ERROR %% default to catch lonely \item

```

\usecounter{CTR} == BEGIN @nmbrlist :=L true
 \@listctr == CTR
 \setcounter{CTR}{0}
END

```

DEFINE \dimen's and \count  
*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

\topsep
\partopsep 1 <*2ekernel>
\itemsep 2 \newskip\topsep
\parsep 3 \newskip\partopsep
\@topsep 4 \newskip\itemsep
\@topsepadd 5 \newskip\parsep
\@outerparskip 6 \newskip\@topsep
7 \newskip\@topsepadd
8 \newskip\@outerparskip

```

(*End of definition for \topsep and others.*)

```

\leftmargin
\rightmargin
\listparindent
\itemindent
\labelwidth
\labelsep
\linewidth
\@totallleftmargin
9 \newdimen\leftmargin
10 \newdimen\rightmargin
11 \newdimen\listparindent
12 \newdimen\itemindent
13 \newdimen\labelwidth
14 \newdimen\labelsep
15 \newdimen\linewidth
16 \newdimen\@totallleftmargin \@totallleftmargin=\z@

```

(*End of definition for \leftmargin and others.*)

```

\leftmargini
\leftmarginii 17 \newdimen\leftmargini
\leftmarginiii 18 \newdimen\leftmarginii
\leftmarginiv 19 \newdimen\leftmarginiii
\leftmarginv 20 \newdimen\leftmarginiv
\leftmarginvi 21 \newdimen\leftmarginv
\leftmarginvii 22 \newdimen\leftmarginvi

(End of definition for \leftmargini and others.)

@listdepth
@itempenalty 23 \newcount\@listdepth \@listdepth=0
@beginparpenalty 24 \newcount\@itempenalty
@endparpenalty 25 \newcount\@beginparpenalty
 26 \newcount\@endparpenalty

(End of definition for \@listdepth and others.)

@labels
 27 \newbox\@labels

(End of definition for \@labels.)

@if@inlabel
@inlabelfalse 28 \newif\if@inlabel \@inlabelfalse
@inlabeltrue
(End of definition for \if@inlabel, @inlabelfalse, and @inlabeltrue.)

@if@newlist
@newlistfalse 29 \newif\if@newlist \@newlistfalse
@newlisttrue
(End of definition for \if@newlist, @newlistfalse, and @newlisttrue.)

@if@noparitem
@noparitemfalse 30 \newif\if@noparitem \@noparitemfalse
@noparitemtrue
(End of definition for \if@noparitem, @noparitemfalse, and @noparitemtrue.)

@if@noparlist
@noparlistfalse 31 \newif\if@noparlist \@noparlistfalse
@noparlisttrue
(End of definition for \if@noparlist, @noparlistfalse, and @noparlisttrue.)

@if@noitemarg
@noitemargfalse 32 \newif\if@noitemarg \@noitemargfalse
@noitemargtrue
(End of definition for \if@noitemarg, @noitemargfalse, and @noitemargtrue.)

@if@newlist
@newlistfalse 33 \newif\if@nmbrlist \@nmbrlistfalse
@newlisttrue
(End of definition for \if@newlist, @newlistfalse, and @newlisttrue.)

```

```

list\list)
 34 \def\list#1#2{%
 35 \ifnum \clistdepth >5\relax
 36 \toodeep
 37 \else
 38 \global\advance\clistdepth\one
 39 \fi
 40 \rightmargin\z@%
 41 \listparindent\z@%
 42 \itemindent\z@%
 43 \csname @list\romannumeral\the\clistdepth\endcsname
 44 \def\@itemlabel{\#1}%
 45 \let\makelabel\mklabel
 46 \nmblistfalse
 47 #2\relax
 48 \trivlist
 49 \parskip\parsep
 50 \parindent\listparindent
 51 \advance\linewidth -\rightmargin
 52 \advance\linewidth -\leftmargin
 53 \advance\@totalleftmargin \leftmargin
 54 \parshape \one \@totalleftmargin \linewidth
 55 \ignorespaces}
(End of definition for \list.)

```

```
\par@deathcycles
 56 \newcount\par@deathcycles
```

(End of definition for \par@deathcycles.)

- \@trivlist** Because \par is sometimes made a no-op it is possible for a missing \item to produce a loop that does not fill memory and so never gets trapped by T<sub>E</sub>X. We thus need to trap this here by setting \par to count the number of times a paragraph ii is called with no progress being made started.

```

 57 \def\@trivlist{%
 58 \if@noskipsec \leavevmode \fi
 59 \topsepadd \topsep
 60 \ifvmode
 61 \advance\topsepadd \partopsep
 62 \else
 63 \unskip \par
 64 \fi
 65 \if@inlabel
 66 \noparitemtrue
 67 \noparlisttrue
 68 \else
 69 \if@newlist \noitemerr \fi
 70 \noparlistfalse
 71 \topsep \topsepadd
 72 \fi
 73 \advance\topsep \parskip
 74 \leftskip \z@skip
 75 \rightskip \rightskip
 76 \parfillskip \flushglue

```

```

77 \par@deathcycles \z@
78 \setpart{if@newlist
79 \advance\par@deathcycles \cne
80 \ifnum \par@deathcycles >\cm
81 \noitemerr
82 {\@par}%
83 \fi
84 \else
85 {\@par}%
86 \fi}%
87 \global \cnewlisttrue
88 \outerparskip \parskip}

```

(End of definition for `\@trivlist`.)

`trivlistlist`)

```

89 \def\trivlist{%
90 \parsep\parskip
91 \cnbrlistfalse
92 \trivlist
93 \labelwidth\z@
94 \leftmargin\z@
95 \itemindent\z@

```

We initialise `\@itemlabel` so that a `trivlist` with an `\item` not having an optional argument doesn't produce an error message.

```

96 \let\@itemlabel\empty
97 \def\makelabel##1{##1}

```

(End of definition for `\trivlist`.)

`\endlist`

```

98 \def\endlist{%
99 \global\advance\clistdepth\mone
100 \endtrivlist}

```

(End of definition for `\endlist`.)

The definition of `\trivlist` used to be in `ltspacedtx` so that other commands could be ‘let to it’. They now use `\def`.

`\endtrivlist`

```

101 \def\endtrivlist{%
102 \if@inlabel
103 \leavevmode
104 \global \cnotinlabelfalse
105 \fi
106 \if@newlist
107 \noitemerr
108 \global \cnewlistfalse
109 \fi
110 \ifhmode\unskip \par

```

We also check if we are in math mode and issue an error message if so (hoping that `\currenvir` resolves suitably). Otherwise the usual “perhaps a missing item” error will get triggered later which is confusing.

```

111 \else

```

```

112 \@inmatherr{\end{\@currenvir}}%
113 \fi
114 \if@nopalst \else
115 \ifdim\lastskip >\z@
116 \tempskipa\lastskip \vskip -\lastskip
117 \advance\tempskipa\parskip \advance\tempskipa -\outerparskip
118 \vskip\tempskipa
119 \fi
120 \endparenv
121 \fi
122 }

```

(End of definition for `\endtrivlist`.)

`\@endparenv` To suppress the paragraph indentation in text immediately following a paragraph-making environment, `\everypar` is changed to remove the space, and `\par` is redefined to restore `\everypar`. Instead of redefining `\par` and `\everypar`, `\@endparenv` was changed to set the `@endpe` switch, letting `\end` redefine `\par` and `\everypar`.

This allows paragraph-making environments to work right when called by other environments. (Changed 27 Oct 86)

In 2024 this logic was partially replaced with a new algorithm:

- `\if@endpe` is now set globally to `true` or `false`.
- In addition `\@endptrue` initiates an `\aftergroup` call to `\propagate@doendpe` if it is used inside a group.
- `\propagate@doendpe` in turn checks the status of `\if@endpe` and if that is `true` it calls `\@doendpe` otherwise it does nothing.
- `\@doendpe` in turn calls `\@endptrue` and also makes the necessary changes to `\par` and `\everypar` so that they handle as before any empty line that follows the environment.
- Because of the `\@endptrue` we get another `\aftergroup`, so the mechanism slowly migrates out of several groups if those follow immediately after the end of the environment. If, however, there is a new paragraph started or an explicit `\par` before the next group ends then this will result in a call to `\@endpfalse` and the migration stops (note that `\propagate@doendpe` is still called once after the group but does nothing).
- Using this approach something like

```

{%
 some customization here
 \begin{equation}
 x=y
 \end{equation}
 some text

```

is still correctly identified as a paragraph continuation so that there is no indentation before `some text`.

- We can get away with using global settings of `\if@endpe` even in nested situations (without keeping track of the status in a stack), because the switch change is made only at the very end of such environments, basically directly before the `\endgroup` in `\end`, and it is later set back to `false` by the next `\everypar` or the next `\par`. Even if the environment is called without using `\begin \end`, the situation doesn't change (or rather cause a problem).
- However, there is one scenario where the new approach would change the behavior. If a box is being built, e.g., with `\setbox`, we have now the case that a `\@endpetrue` inside would migrate out into a context in which it should not be true (because the box might get used elsewhere, e.g., a float). In the past, due to local switch changes, that didn't happen, i.e., `\if@endpe` would revert to `false` at the end of the box definition.
- Thus, to avoid that one has to explicitly set it back to false at the end of such constructions, just as we also need to prevent colors from migrating out. Thus the correct place to do this is in `\color@endgroup` because that is always called at that point.

```

123 \def\@endparenv{%
124 \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}
125 <latexrelease>\IncludeInRelease{2015/01/01}{\@doendpe}{clubpenalty fix}%
126 \def\@doendpe{\@endpetrue
127 \def\par{\@restorepar}
```

If a section heading changes `\clubpenalty` to keep lines after it together then this modification is restored via the `\everypar` mechanism at the start of the next paragraph. As we destroy the contents of this token here we explicitly set `\clubpenalty` back to its default.

```

128 \clubpenalty\@clubpenalty
129 \@endpefalse
130 \everypar{}{\par}%

```

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).

```

131 \everypar
132 {\{\setbox\z@\lastbox\}%
133 \everypar{}{\@endpefalse}}
134 <latexrelease>\EndIncludeInRelease
135 <latexrelease>\IncludeInRelease{0000/00/00}{\@doendpe}{clubpenalty fix}%
136 <latexrelease>\def\@doendpe{\@endpetrue
137 <latexrelease> \def\par{\@restorepar\everypar{}{\par\@endpefalse}\everypar
138 <latexrelease> {\{\setbox\z@\lastbox\}\everypar{}{\@endpefalse}}
139 <latexrelease>\EndIncludeInRelease
```

*(End of definition for `\@endparenv` and `\@doendpe`.)*

|                                                                                                                    |                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\if@endpe</code><br><code>\@endpefalse</code><br><code>\@endpetrue</code><br><code>\propagate@doendpe</code> | As outlined above these are no longer simple switches, but we keep the name because it is used all over the place.<br><br><code>\newif\if@endpe</code> |
|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|

```

141 </2ekernel>
142 <*2ekernel | latexrelease>
143 <latexrelease>\IncludeInRelease{2024/11/01}%
144 <latexrelease> {\@endpefalse}{New @endpe handling}%
145 \def\@endpefalse{\global\let\if@endpe\iffalse}
146 \def\@endpetrue {%
147 \global\let\if@endpe\iftrue

```

If we are inside a group then propagate to the outside:

```

148 \ifnum\currentgrouplevel>\z@%
149 \aftergroup\propagate@doendpe
150 \fi
151 }

```

If `\if@endpe` is still true after the group ends, we run `\@doendpe` that in turn runs another `\@endpetrue` (besides other things), thus propagating further if necessary. However, if the endpe situation got resolved and `\if@endpe` is `false` then nothing further happens.

```

152 \def\propagate@doendpe{\if@endpe \@doendpe \fi}
153 </2ekernel | latexrelease>
154 <latexrelease>\EndIncludeInRelease
155 <latexrelease>\IncludeInRelease{0000/00/00}%
156 <latexrelease> {\@endpetrue}{New @endpe handling}%
157 <latexrelease>
158 <latexrelease>
159 <latexrelease>\def\@endpefalse{\let\if@endpe\iffalse}
160 <latexrelease>\def\@endpetrue{\let\if@endpe\iftrue}
161 <latexrelease>
162 <latexrelease>\EndIncludeInRelease
163 <*2ekernel>

```

*(End of definition for `\if@endpe` and others.)*

```
\@mklab
164 \def\@mklab#1{\hfil #1}
```

*(End of definition for `\@mklab`.)*

```
\item
165 \def\item{%
166 \inmatherr\item
167 \ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

*(End of definition for `\item`.)*

```
\@donoparitem
168 \def\@donoparitem{%
169 \donoparitemfalse
170 \global\setbox\@labels\hbox{\hskip -\leftmargin
171 \unhbox\@labels
172 \hskip \leftmargin}%
173 \if@minipage\else
174 \tempskip\lastskip
175 \vskip -\lastskip
176 \advance\tempskip\outerparskip
```

```

177 \advance\@tempskipa -\parskip
178 \vskip\@tempskipa
179 \fi}

```

(End of definition for `\@donoparitem`.)

```

\@item
180 \def\@item[#1]{%
181 \if@noperitem
182 \@donoparitem
183 \else
184 \if@inlabel
185 \indent \par
186 \fi
187 \ifhmode
188 \unskip\unskip \par
189 \fi
190 \if@newlist
191 \if@nobreak
192 \@nbitem
193 \else
194 \addpenalty\@beginparpenalty
195 \addvspace\@topsep
196 \addvspace{-\parskip}%
197 \fi
198 \else
199 \addpenalty\@itempenalty
200 \addvspace\itemsep
201 \fi
202 \global\@inlabeltrue
203 \fi
204 \everypar{%
205 \minipagetrue
206 \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.

```

207 \if@inlabel
208 \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.

```

209 {\setbox\z@\lastbox
210 \ifvoid\z@
211 \kern-\itemindent
212 \fi}%
213 \box\@labels
214 \penalty\z@
215 \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.

```

216 \if@nobreak
217 \nobreakfalse
218 \clubpenalty \OM
219 \else
220 \clubpenalty \clubpenalty
221 \everypar{}%
222 \fi}%
223 \if@noitemarg
224 \noitemargfalse
225 \if@nbrlist
226 \refstepcounter\listctr
227 \fi
228 \fi

```

We use \sbox to support colour commands.

```

229 \sbox\tempboxa{\makelabel{#1}}%
230 \global\setbox\labels\hbox{%
231 \unhbox\labels
232 \hskip \itemindent
233 \hskip -\labelwidth
234 \hskip -\labelsep
235 \ifdim \wd\tempboxa >\labelwidth
236 \box\tempboxa
237 \else
238 \hbox to\labelwidth {\unhbox\tempboxa}%
239 \fi
240 \hskip \labelsep}%
241 \ignorespaces}

```

(End of definition for \item.)

```
\makelabel
242 \def\makelabel#1{%
243 \if@latex@error{Lonely \string\item--perhaps a missing
244 list environment}\@ehc}

```

(End of definition for \makelabel.)

```
\@nbitem
245 \def\@nbitem{%
246 \tempskipa\outerparskip
247 \advance\tempskipa -\parskip
248 \addvspace\tempskipa}

```

(End of definition for \nbitem.)

```
\usecounter
249 \def\usecounter#1{\@nmbrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}
(End of definition for \usecounter.)
```

## 1.6 Itemize and Enumerate

Enumeration is done with four counters: `enumi`, `enumii`, `enumiii` and `enumiv`, where `enumN` controls the numbering of the Nth level enumeration. The label is generated by the commands `\labelenumi` ... `\labelenumiv`, which should be defined by the document style. Note that `\p@enumN\theenumN` defines the output of a `\ref` command. A typical definition might be:

```
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}
\def\labelenumii{(\theenumii)}
```

which will print the labels as ‘(a)’, ‘(b)’, ... and print a `\ref` as ‘3a’.

The item numbers are moved to the right of the label box, so they are always a distance of `\labelsep` from the item.

`\@enumdepth` holds the current enumeration nesting depth.

Itemization is controlled by four commands: `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`. To cause the second-level list to be bulleted, you just define `\labelitemii` to be `•`. `\@itemspacing` and `\@itemdepth` are the analogs of `\@enumspacing` and `\@enumdepth`.

*Historical L<sup>A</sup>T<sub>E</sub>X 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
250 \newcount\@enumdepth \@enumdepth = 0
(End of definition for \@enumdepth.)
```

```
\c@enumi
\c@enumii 251 \def\c@enumii{\@definecounter{enumii}}
\c@enumiii 252 \def\c@enumiii{\@definecounter{enumiii}}
\c@enumiv 253 \def\c@enumiv{\@definecounter{enumiv}}
```

(End of definition for `\c@enumi` and others.)

`enumerate (env.)`

```
255 \def\enumerate{%
256 \ifnum \c@enumdepth > \thr@@\c@toodeep\else
257 \advance\c@enumdepth\@ne
258 \edef\c@enumctr{\enum\romannumeral\the\c@enumdepth}%
259
260 \expandafter
261 \list
262 \csname label\c@enumctr\endcsname
263 {\usecounter\c@enumctr\def\makelabel##1{\hss\llap{##1}}}\%
264 \fi}
264 \let\endenumerate =\endlist
```

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
\itemize ==
BEGIN
 if \c@itemdepth > 3
 then errormessage: 'Too deeply nested'.
 else \c@itemdepth := \c@itemdepth + 1
 \c@itemitem == eval(labelitem\romannumeral\the\c@itemdepth)
 \list{\c@nameuse{\c@itemitem}}
 {\makelabel{LABEL} == \hss \llap{LABEL}}
 fi
END

\enditemize == \endlist
```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

`\c@itemdepth`

```
265 \newcount\c@itemdepth \c@itemdepth = 0
```

(End of definition for `\c@itemdepth`.)

`itemize (env.)`

```
266 \def\itemize{%
267 \ifnum \c@itemdepth > \thr@@\c@toodeep\else
268 \advance\c@itemdepth\@ne
269 \edef\c@itemitem{\labelitem\romannumeral\the\c@itemdepth}%
270
271 \expandafter
272 \list
273 \csname\c@itemitem\endcsname
274 {\def\makelabel##1{\hss\llap{##1}}}\%
274 \fi}
275 \let\enditemize =\endlist
276
```

# File 40

## ltboxes.dtx

### 1 L<sup>A</sup>T<sub>E</sub>X 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 T<sub>E</sub>X box register.

\savebox \savebox{⟨cmd⟩} ... : ⟨cmd⟩ is defined to be a T<sub>E</sub>X 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 (env.) \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
\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 (env.)` `\minipage` : Similar to `\parbox`, except it also makes this look like a page by setting `\textwidth == \columnwidth == box width` changes footnotes by redefining:

```

\@mpfn == mpfootnote
\thempfn == \thempfootnote
\@footnotetext == \@mpfootnotetext

```

resets the following list environment parameters:

```

\@listdepth == \@mplistdepth

```

where `\@mplistdepth` is initialized to zero,

and executes `\minipagerestore` to allow the document style to reset any other parameters it desires. It sets `@minipage` true, and resets `\everypar` to set it false. This switch keeps `\addvspace` from putting space at the top of a minipage.

Change added 24 May 89: `\minipage` sets `@minipage` globally; `\endminipage` resets it false.

`\rule`      `\rule[<raised>]{<width>}{<height>}` : Makes a `<width> * <height>` rule, raised `<raised>`.

`\underline`    `\underline{<text>}` : Makes an underlined hbox with `<text>` in it.

`\raisebox`    `\raisebox{<distance>}[<height>][<depth>]{<box>}` :

Raises `<box>` up by `<distance>` length (down if `<distance>` negative). Makes `TEX` think that the new box extends `<height>` above the line and `<depth>` below, for a total vertical length of `<height>+<depth>`. Default values of `<height>` & `<depth>` = actual height and depth of box in new position.

```

1 {*2ekernel}
2 \message{boxes,}

```

`\makebox`    `\makebox` User level command just looks for optional [ or (.

```

3 {/2ekernel}
4 {(\@latexrelease)\IncludeInRelease{2015/01/01}\%}
5 {(\@latexrelease)} {\@makebox}{\Make\makebox robust}\%
6 {(*2ekernel|\@latexrelease)}
7 \DeclareRobustCommand\makebox{\%
8 \leavevmode
9 \@ifnextchar(%)
10 \makepicbox
11 {\ifnextchar[\@makebox\mbox}\%
12 {(/2ekernel|\@latexrelease)}
13 {(\@latexrelease)\EndIncludeInRelease}

```

```

14 <latexrelease>\IncludeInRelease{0000/00/00}%
15 <latexrelease> {\makebox}{\Make \makebox robust}%
16 <latexrelease>\def\makebox{%
17 <latexrelease> \leavevmode
18 <latexrelease> \@ifnextchar(%)
19 <latexrelease> \makepicbox
20 <latexrelease> {\ifnextchar[\@makebox\mbox}%
21 <latexrelease>\expandafter\let\csname makebox \endcsname\@undefined
22 <latexrelease>\EndIncludeInRelease
23 {*2ekernel}

```

(End of definition for `\makebox`.)

`\mbox` The basic horizontal box command for L<sup>A</sup>T<sub>E</sub>X.

```
24 \DeclareRobustCommand{\mbox}[1]{\leavevmode\hbox{#1}}
```

(End of definition for `\mbox`.)

`\@makebox` Look for a possible second optional argument (defaults to `c`).

```

25 \def\@makebox[#1]{%
26 \ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}

```

(End of definition for `\@makebox`.)

`\@begin@tempboxa` Helper macro for supporting `\height`, `\width` etc. Grab #1 into `\@tempboxa` and measure it.

```

27 \long\def\@begin@tempboxa#1#2{%
28 \begingroup
29 \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
30 \def\width{\wd\@tempboxa}%
31 \def\height{\ht\@tempboxa}%
32 \def\depth{\dp\@tempboxa}%
33 \let\totalheight\ovr@i
34 \totalheight\height
35 \advance\totalheight\depth}

```

(End of definition for `\@begin@tempboxa`.)

`\@end@tempboxa` End the group started by `\@begin@tempboxa`, so that the scope of `\height` only includes the ‘length’ argument to the user-command.

```
36 \let\@end@tempboxa\endgroup
```

(End of definition for `\@end@tempboxa`.)

`\bm@c` Set up spacing.

```

37 \def\bm@c{\hss\unhbox\@tempboxa\hss}
38 \def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
39 \def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
40 \def\bm@s{\unhbox\@tempboxa}

```

(End of definition for `\bm@c` and others.)

\@imakebox Internal form of \makebox.

```
41 〈/2ekernel〉
42 〈latexrelease〉\IncludeInRelease{2023/11/01}%
43 〈latexrelease〉 {〈@imakebox〉{Unknown alignment warning}%
44 〈*2ekernel | latexrelease〉
45 \long\def\@imakebox[#1] [#2]#3{%
46 \begin{tempboxa}\hbox{#3}%
47 \setlength\tempdima{#1}%
48 support calc
49 \hb@xt@\tempdima{%
50 \expandafter\ifx\csname bm@#2\endcsname\relax
51 \bm@c
52 \@latex@warning{Unexpected alignment #2}%
53 \else
54 \csname bm@#2\endcsname
55 \fi}%
56 \end{tempboxa}%
57 〈/2ekernel | latexrelease〉
58 〈latexrelease〉\EndIncludeInRelease
59 〈latexrelease〉\IncludeInRelease{0000/00/00}%
60 〈latexrelease〉 {〈@imakebox〉{Unknown alignment warning}%
61 〈latexrelease〉\long\def\@imakebox[#1] [#2]#3{%
62 \begin{tempboxa}\hbox{#3}%
63 \setlength\tempdima{#1}%
64 support calc
65 \hb@xt@\tempdima{\csname bm@#2\endcsname}%
66 \end{tempboxa}%
67 〈/2ekernel | latexrelease〉
68 〈/2ekernel〉
```

(End of definition for \@imakebox.)

\@makepicbox Picture mode form of \makebox.

```
67 \def\@makepicbox(#1,#2){%
68 \ifnextchar[{\@imakepicbox(#1,#2)}{\@imakepicbox(#1,#2)[]}}
```

(End of definition for \@makepicbox.)

\@imakepicbox picture mode version

```
69 〈/2ekernel〉
70 〈*2ekernel | latexrelease〉
71 〈latexrelease〉\IncludeInRelease{2020/10/01}%
72 〈latexrelease〉 {〈@imakepicbox〉{default units}%
73 \long\def\@imakepicbox(#1,#2)[#3]#4{%
74 \defaultunits\tempdimc{#2}\unitlength
75 \vbox to\tempdimc{#2}{%
76 \let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
77 \let\mb@t\vss
78 \tfor\reserved@a :=#3\do{%
79 \if s\reserved@a
80 \let\mb@l\relax\let\mb@r\relax
81 \else
82 \expandafter\let\csname mb@\reserved@a\endcsname\relax
83 \fi}%
84 \mb@t
85 \defaultunits\tempdimc{#1}\unitlength}
```

```

86 \hb@xt@{\tempdimc{\mb@l #4\mb@r}}%
87 \mb@b
88 \kern{z@}}
89
```

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 L<sup>A</sup>T<sub>E</sub>X Book. The kern is removed in compatibility mode.

```

90
```

`\kern{z@}`

```

91
```

`\EndIncludeInRelease`

```

92
```

`\IncludeInRelease{0000/00/00}%`

```

93
```

`\@imakepicbox{default units}%`

```

94
```

`\long\def\@imakepicbox(#1,#2)[#3]#4{%`

```

95
```

`\vbox to#2\unitlength`

```

96
```

`{\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss`

```

97
```

`\let\mb@t\vss`

```

98
```

`\@tfor\reserved@a :=#3\do{%`

```

99
```

`\if s\reserved@a`

```

100
```

`\let\mb@l\relax\let\mb@r\relax`

```

101
```

`\else`

```

102
```

`\expandafter\let\csname mb@\reserved@a\endcsname\relax`

```

103
```

`\fi}%`

```

104
```

`\mb@t`

```

105
```

`\hb@xt#1\unitlength{\mb@l #4\mb@r}%`

```

106
```

`\mb@b`

```

107
```

`\kern{z@}}`

```

108
```

`\EndIncludeInRelease`

```

109
```

`(*2ekernel)`

(End of definition for `\@imakepicbox`.)

`\set@color` This macro is initially a no-op, but the color package will redefine it to insert a `\special`.

```

109
```

`\let\set@color\relax`

(End of definition for `\set@color`.)

`\color@begingroup` In the past these macros were initially no-ops, and the color package redefined them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`, `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and `(set to main document color)` respectively.

Nowadays we always set the group already in the kernel as this makes the coding simpler.

```

110
```

`(*2ekernel | latexrelease)`

```

111
```

`\EndIncludeInRelease{2021/06/01}%`

```

112
```

`\@imakepicbox{color group settings}%`

```

113
```

`\let\color@begingroup\begingroup`

```

114
```

`\def\color@setgroup{\color@begingroup} % changed further in color package`

```

115
```

`\let\normalcolor\relax % remains untouched; only changed in a color pa`

```

116
```

`\def\color@hbox{\hbox\bgroup\color@begingroup}`

```

117
```

`\def\color@vbox{\vbox\bgroup\color@begingroup}`

```

118
```

`\def\color@endbox{\color@endgroup\egroup}`

```

119
```

`\EndIncludeInRelease`

```

120
```

```

122 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
123 〈\latexrelease〉 {\color@begingroup}{color group settings}%
124 〈\latexrelease〉
125 〈\latexrelease〉\let\color@begingroup\relax
126 〈\latexrelease〉\let\color@setgroup\relax
127 〈\latexrelease〉\let\normalcolor\relax
128 〈\latexrelease〉\let\color@hbox\relax
129 〈\latexrelease〉\let\color@vbox\relax
130 〈\latexrelease〉\let\color@endbox\relax
131 〈\latexrelease〉
132 〈\latexrelease〉\EndIncludeInRelease
133 〈*2ekernel〉

```

(End of definition for `\color@begingroup` and others.)

`\color@endgroup` This macro is separated out because it received an update in 2024, so requires its own rollback.

```

134 〈/2ekernel〉
135 〈*2ekernel | \latexrelease〉
136 〈\latexrelease〉\IncludeInRelease{2024/11/01}%
137 〈\latexrelease〉 {\color@endgroup}{color group settings}%

```

Beside `\endgraf` for handling vertical boxes we also reset `\if@endpe` as we are leaving the context.

```

138 \def\color@endgroup{\endgraf\@endpefalse\endgroup}
139 〈\latexrelease〉
140 〈/2ekernel | \latexrelease〉
141 〈\latexrelease〉\EndIncludeInRelease

142 〈\latexrelease〉\IncludeInRelease{2021/06/01}%
143 〈\latexrelease〉 {\color@endgroup}{color group settings}%
144 〈\latexrelease〉\def\color@endgroup{\endgraf\endgroup}
145 〈\latexrelease〉
146 〈\latexrelease〉\EndIncludeInRelease

147 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
148 〈\latexrelease〉 {\color@endgroup}{color group settings}%
149 〈\latexrelease〉
150 〈\latexrelease〉\let\color@endgroup\relax
151 〈\latexrelease〉
152 〈\latexrelease〉\EndIncludeInRelease
153 〈*2ekernel〉

```

(End of definition for `\color@endgroup`.)

`\newsavebox` Allocate a new ‘savebox’.

```
154 \def\newsavebox#1{\@if definable{#1}{\newbox#1}}
```

(End of definition for `\newsavebox`.)

`\savebox` Save #1 in a box register.

```

155 〈/2ekernel〉
156 〈\latexrelease〉\IncludeInRelease{2015/01/01}%
157 〈\latexrelease〉 {\savebox}{Make \savebox robust}%
158 〈*2ekernel | \latexrelease〉
159 \DeclareRobustCommand\savebox[1]{%

```

```

160 \@ifnextchar(%)
161 {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}}%
162 </2ekernel | latexrelease>
163 <latexrelease>\EndIncludeInRelease
164 <latexrelease>\IncludeInRelease{0000/00/00}%
165 <latexrelease> {\@savebox}{\Make \savebox robust}%
166 <latexrelease>\def\savebox#1{%
167 <latexrelease> \@ifnextchar(%)
168 <latexrelease> {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}}%
169 <latexrelease>\expandafter\let\csname savebox \endcsname\undefined
170 <latexrelease>\EndIncludeInRelease
171 <*2ekernel>

```

(End of definition for `\savebox`.)

`\sbox` Save #1 in a box register.

```

172 \DeclareRobustCommand\sbox[2]{\setbox#1\hbox{%
173 \color@setgroup#2\color@endgroup}}

```

(End of definition for `\sbox`.)

`\@savebox` Look for second optional argument.

```

174 \def\@savebox#1[#2]{%
175 \@ifnextchar [{\@isavebox#1[#2]}{\@isavebox#1[#2][c]}}

```

(End of definition for `\@savebox`.)

`\@isavebox`

```

176 \long\def\@isavebox#1[#2][#3][#4]{%
177 \sbox#1{\imakebox[#2][#3][#4]}}

```

(End of definition for `\@isavebox`.)

`\@savepicbox` Picture mode version of `\savebox`.

```

178 \def\@savepicbox#1(#2,#3){%
179 \@ifnextchar[%]
180 {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3)[]}}

```

(End of definition for `\@savepicbox`.)

`\@isavepicbox` Picture mode version of `\savebox`.

```

181 \long\def\@isavepicbox#1(#2,#3)[#4][#5]{%
182 \sbox#1{\imakepicbox(#2,#3)[#4][#5]}}

```

(End of definition for `\@isavepicbox`.)

`\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.

```

183 \def\lrbox#1{%
184 \edef\reserved@a{%
185 \endgroup
186 \setbox#1\hbox{%
187 \begingroup\aftergroup}%
188 \def\noexpand\@currenvir{\currenvir}%

```

```

189 \def\noexpand\@currenvline{\on@line}%
190 \reserved@a
191 \endpfalse
192 \color@setgroup
193 \ignorespaces}

(End of definition for \lrbox.)

\endlrbox End the lrbox environment.
194 \def\endlrbox{\unskip\color@endgroup}

(End of definition for \endlrbox.)

\usebox unchanged
195 \DeclareRobustCommand\usebox[1]{\leavevmode\copy #1\relax}

(End of definition for \usebox.)

\fbox The following definition of \fbox was written by Pavel Curtis (Extra space removed 14
Jan 88) RmS 92/08/24: Replaced occurrence of \halfwidth by \wholewidth
196 \DeclareRobustCommand\fbox[1]{%
197 \leavevmode
198 \hbox{%
199 \hskip-\wholewidth
200 \vbox{%
201 \vskip-\wholewidth
202 \hrule \height\wholewidth
203 \hbox{%
204 \vrule\width\wholewidth
205 #1%
206 \vrule\width\wholewidth}%
207 \hrule\height\wholewidth
208 \vskip-\wholewidth}%
209 \hskip-\wholewidth}}
}

(End of definition for \fbox.)

\fboxrule user level parameters,
\fboxsep 210 \newdimen\fboxrule
211 \newdimen\fboxsep

(End of definition for \fboxrule and \fboxsep.)

\fbox Abbreviated framed box command.
212 \DeclareRobustCommand\fbox[1]{%
213 \leavevmode
214 \setbox\tempboxa\hbox{%
215 \color@begingroup
216 \kern\fboxsep{#1}\kern\fboxsep
217 \color@endgroup}%
218 \framebx\relax}

(End of definition for \fbox.)

```

\framebox Framed version of \makebox.

```

219 </2ekernel>
220 <latexrelease>\IncludeInRelease{2015/01/01}%
221 <latexrelease> {\framebox}{\Make \framebox robust}%
222 <*2ekernel | latexrelease>
223 \DeclareRobustCommand\framebox{%
224 \@ifnextchar(%)
225 \@framepicbox{\@ifnextchar[\@framebox\fbox}{}%
226 </2ekernel | latexrelease>
227 <latexrelease>\EndIncludeInRelease
228 <latexrelease>\IncludeInRelease{0000/00/00}%
229 <latexrelease> {\framebox}{\Make \framebox robust}%
230 <latexrelease>\def\framebox{%
231 \@ifnextchar(%)
232 \@framepicbox{\@ifnextchar[\@framebox\fbox}{}%
233 <latexrelease>\expandafter\let\csname framebox \endcsname\@undefined
234 <latexrelease>\EndIncludeInRelease
235 <*2ekernel>

```

(End of definition for \framebox.)

\@framebox Deal with optional arguments.

```

236 \def\@framebox[#1]{%
237 \@ifnextchar[%
238 {\@ifframebox[#1]}%
239 {\@ifframebox[#1][c]}}

```

(End of definition for \@framebox.)

\@ifframebox The handling the optional arguments. In order to set the whole box, including the frame to the specified dimension, we first determine that dimension from the natural size of the text, #3. calculated width.

```

240 </2ekernel>
241 <latexrelease>\IncludeInRelease{2023/11/01}%
242 <latexrelease> {\@ifframebox}{\Unknown alignment warning}%
243 <*2ekernel | latexrelease>
244 \long\def\@ifframebox[#1][#2]{#3}%
245 \leavevmode
246 \@begin@tempboxa\hbox{#3}%
247 \setlength\tempdima{#1}%
248 \setbox\tempboxa\hb@xt@\tempdima
249 {\kern\fboxsep
250 \expandafter\ifx\csname bm@#2\endcsname\relax
251 \bm@c
252 \@latex@warning{Unexpected alignment #2}%
253 \else
254 \csname bm@#2\endcsname
255 \fi
256 \kern\fboxsep}%
257 \@frameb@x{\kern\fboxrule}%
258 \@end@tempboxa}
259 </2ekernel | latexrelease>
260 <latexrelease>\EndIncludeInRelease
261 <latexrelease>\IncludeInRelease{0000/00/00}%

```

```

262 <|latexrelease> {\@ifframebox}{Unknown alignment warning}%
263 <|latexrelease>\long\def\@ifframebox[#1][#2]{%
264 <|latexrelease> \leavevmode
265 <|latexrelease> \begin{tempboxa}\hbox{#3}%
266 <|latexrelease> \setlength{\tempdima}{#1}%
267 <|latexrelease> \setbox{\tempboxa}\hb@xt@{\tempdima}%
268 <|latexrelease> { \kern\fboxsep\csname bm@\#2\endcsname\kern\fboxsep}%
269 <|latexrelease> \frameb@x{\kern-\fboxrule}%
270 <|latexrelease> \end{tempboxa}%
271 <|latexrelease>\EndIncludeInRelease
272 {*2ekernel}

```

(End of definition for `\@ifframebox`.)

- `\@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.

```

273 \def\@frameb@x#1{%
274 \tempdima\fboxrule
275 \advance\tempdima\fboxsep
276 \advance\tempdima\dp\tempboxa
277 \hbox{%
278 \lower\tempdima\hbox{%
279 \vbox{%
280 \hrule\height\fboxrule
281 \hbox{%
282 \vrule\width\fboxrule
283 #1%
284 \vbox{%
285 \vskip\fboxsep
286 \box\tempboxa
287 \vskip\fboxsep}%
288 #1%
289 \vrule\width\fboxrule}%
290 \hrule\height\fboxrule}%
291 }%
292 }%
293 }

```

(End of definition for `\@frameb@x`.)

- `\@framepicbox` Picture mode version.

```

294 \def\@framepicbox(#1,#2){%
295 \@ifnextchar[\{\@framepicbox(#1,#2)\}{\@framepicbox(#1,#2)[]}}

```

(End of definition for `\@framepicbox`.)

- `\@ifframepicbox` Picture mode version.

```

296 \long\def\@ifframepicbox(#1,#2)[#3]{%
297 \frame{\@imakepicbox(#1,#2)[#3]{#4}}}

```

(End of definition for `\@ifframepicbox`.)

\parbox The main vertical-box command for L<sup>A</sup>T<sub>E</sub>X.

```

298 {/2ekernel}
299 <|latexrelease>\IncludeInRelease{2015/01/01}%
300 <|latexrelease> {\parbox}{\Make \parbox robust}%
301 <*2ekernel | latexrelease>
302 \DeclareRobustCommand\parbox{%
303 \@ifnextchar[%
304 \@iparbox
305 {\@iiiparbox c\relax[s]}%
306 {/2ekernel | latexrelease}
307 <|latexrelease>\EndIncludeInRelease
308 <|latexrelease>\IncludeInRelease{0000/00/00}%
309 <|latexrelease> {\parbox}{\Make \parbox robust}%
310 <|latexrelease>\def\parbox{%
311 <|latexrelease> \@ifnextchar[%
312 <|latexrelease> \@iparbox
313 <|latexrelease> {\@iiiparbox c\relax[s]}%
314 <|latexrelease>\expandafter\let\csname parbox \endcsname\@undefined
315 <|latexrelease>\EndIncludeInRelease
316 <*2ekernel>

```

(End of definition for \parbox.)

\@iparbox Optional argument handling.

```

317 \def\@iparbox[#1]{%
318 \@ifnextchar[%
319 {\@iiiparbox[#1]}%
320 {\@iiiparbox[#1]\relax[s]}%

```

(End of definition for \@iparbox.)

\@iiiparbox Optional argument handling.

```

321 \def\@iiiparbox#1[#2]{%
322 \@ifnextchar[%
323 {\@iiiparbox[#1]{#2}}%
324 {\@iiiparbox[#1]{#2}[#1]}%

```

(End of definition for \@iiiparbox.)

\@iiiparbox The internal version of \parbox.

\@parboxto

```

325 \let\@parboxto\@empty
326 {/2ekernel}
327 <|latexrelease>\IncludeInRelease{2023/11/01}%
328 <|latexrelease> {\@iiiparbox}{Unknown alignment warning}%
329 <*2ekernel | latexrelease>
330 \long\def\@iiiparbox#1#2[#3]#4#5{%
331 \leavevmode
332 \@pboxswfalse
333 \setlength\@tempdima{#4}%
334 \begin{\@tempboxa}\vbox{\hsize\@tempdima\@parboxrestore#5\@par}%
335 \ifx\relax#2\else
336 \setlength\@tempdimb{#2}%
337 \edef\@parboxto{to\the\@tempdimb}%
338 \fi

```

```

339 \if#1b\vbox
340 \else\if #1t\vtop
341 \else\ifmmode\vcenter
342 \else\@pboxswtrue \$\vcenter
343 \fi\fi\fi
344 \parboxto{\let\hss\vss\let\unhbox\unvbox
345 \expandafter\ifx\csname bm@#3\endcsname\relax
346 \bm@c
347 \@latex@warning{Unexpected alignment #3}%
348 \else
349 \csname bm@#3\endcsname
350 \fi}%
351 \if@pboxsw \m@th$\fi
352 \end@tempboxa}
353 </2ekernel | latexrelease>
354 <latexrelease>\EndIncludeInRelease
355 <latexrelease>\IncludeInRelease{0000/00/00}%
356 <latexrelease> {\@iiiparbox}{Unknown alignment warning}%
357 <latexrelease>\long\def\@iiiparbox#1#2[#3]#4#5{%
358 <latexrelease> \leavevmode
359 <latexrelease> \@pboxswfalse
360 <latexrelease> \setlength\@tempdima{#4}%
361 <latexrelease> \begin@tempboxa\vbox{\hsize\@tempdima\parboxrestore#5\@par}%
362 <latexrelease> \ifx\relax#2\else
363 <latexrelease> \setlength\@tempdimb{#2}%
364 <latexrelease> \edef\@parboxto{to\the\@tempdimb}%
365 <latexrelease> \fi
366 <latexrelease> \if#1b\vbox
367 <latexrelease> \else\if #1t\vtop
368 <latexrelease> \else\ifmmode\vcenter
369 <latexrelease> \else\@pboxswtrue \$\vcenter
370 <latexrelease> \fi\fi\fi
371 <latexrelease> \parboxto{\let\hss\vss\let\unhbox\unvbox
372 <latexrelease> \csname bm@#3\endcsname}%
373 <latexrelease> \if@pboxsw \m@th$\fi
374 <latexrelease> \end@tempboxa}
375 <latexrelease>\EndIncludeInRelease
376 <*2ekernel>

```

(End of definition for `\@iiiparbox` and `\@parboxto`.)

`\@arrayparboxrestore` Restore various paragraph parameters.

The rational for allowing two normally global flags to be set locally here 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 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 `\@setnobreak`; otherwise this command will be redundant.

```

377 </2ekernel>
378 <latexrelease>\IncludeInRelease{2017-04-15}%
379 <latexrelease> {\normalallineskiplimit}%
380 <latexrelease> {reset \lineskiplimit}%
381 <*2ekernel | latexrelease>

```

```

382 \def\@arrayparboxrestore{%
383 \let\ifnobreak\iffalse
384 \let\ifnoskipsec\iffalse
385 \let\par\@@par
386 \let\-\@dischyp
Redefined accents to allow changes in font encoding
387 \let\'\@acci\let`\@accii\let\=\@acciii
388 \parindent\z@\parskip\z@skip
389 \everypar{}%
390 \linewidth\hsize
391 \totalleftmargin\z@
392 \leftskip\z@skip \rightskip\z@skip \rightskip\z@skip
393 \parfillskip\@flushglue
394 \lineskip\normallineskip
395 \lineskiplimit\normallineskiplimit
396 \baselineskip\normalbaselineskip
397 \sloppy}
398 {/2ekernel | latexrelease}

399 <| latexrelease>\EndIncludeInRelease
400 <| latexrelease>\IncludeInRelease{0000-00-00}%
401 <| latexrelease> {\normallineskiplimit}
402 <| latexrelease> {reset \lineskiplimit}%
403 <| latexrelease>\def\@arrayparboxrestore{%
404 \let\ifnobreak\iffalse
405 \let\ifnoskipsec\iffalse
406 \let\par\@@par
407 \let\-\@dischyp
408 \let\'\@acci\let`\@accii\let\=\@acciii
409 \parindent\z@\parskip\z@skip
410 \everypar{}%
411 \linewidth\hsize
412 \totalleftmargin\z@
413 \leftskip\z@skip \rightskip\z@skip \rightskip\z@skip
414 \parfillskip\@flushglue \lineskip\normallineskip
415 \baselineskip\normalbaselineskip
416 \sloppy}
417 <| latexrelease>\EndIncludeInRelease
418 {*2ekernel}

(End of definition for \@arrayparboxrestore.)

```

**\@parboxrestore** Restore various paragraph parameters, and also \\.

```

419 \def\@parboxrestore{\@arrayparboxrestore\let\\\\@normalcr}

```

(End of definition for \@parboxrestore.)

**\if@minipage** Switch that is true at the start of a minipage.

```

420 \def\@minipagefalse{\global\let\if@minipage\iffalse}
421 \def\@minipagetrue {\global\let\if@minipage\iftrue}
422 \f@minipagefalse

```

(End of definition for \if@minipage.)

```

\if@in@minipage@env
 423 \newif\if@in@minipage@env
 (End of definition for \if@in@minipage@env.)
```

\minipage Essentially an environment form of \parbox.

```

 424 \def\minipage{%
 425 \@ifnextchar[%]
 426 \@iminiplate
 427 {\@iiiminipage c\relax[s]}}
```

(End of definition for \minipage.)

\@iminiplate Optional argument handling.

```

 428 \def\@iminiplate[#1]{%
 429 \@ifnextchar[%]
 430 {\@iiiminipage{#1}}%
 431 {\@iiiminipage{#1}\relax[s]}}
```

(End of definition for \@iminiplate.)

\@iiiminipage Optional argument handling.

```

 432 \def\@iiiminipage[#1][#2]{%
 433 \@ifnextchar[%]
 434 {\@iiiminipage[#1]{#2}}%
 435 {\@iiiminipage[#1]{#2}[#1]}}
```

(End of definition for \@iiiminipage.)

\@iiiminipage Internal form of minipage.

```

 436 \def\@iiiminipage#1#2[#3]{%
 437 \leavevmode
 438 \pboxswfalse
 439 \setlength\tempdima{#4}%
 440 \def\@mpargs{{#1}{#2}[#3]{#4}}%
 441 \setbox\tempboxa\vbox\bgroup
 442 \color@begingroup
 443 \hsize\tempdima
 444 \textwidth\hsize \columnwidth\hsize}
```

We check for nested minipages inside the box so that there is always a group resetting the switch even if the code does not use \begin to start the minipage.

```

 445 \if@in@minipage@env
```

We only issue a warning if the outer minipage contained footnotes because that is the problematical case.

```

 446 \ifvoid\@mpfootins\else
 447 \@latex@warning{Nested minipage:
 448 footnotes may be misplaced}%
 449 \fi
 450 \else
 451 \Gin@minipage@envtrue
 452 \fi
```

```

453 \parboxrestore
454 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
455 \let\@footnotetext\@mpfootnotetext
456 \let\@listdepth\@mplistdepth \c@mplistdepth\z@
457 \minipagerestore
458 \c@setminipage}

(End of definition for \c@iiminipage.)

\@minipagerestore Hook so that other styles can reset other commands in a minipage.
459 \let\@minipagerestore=\relax

(End of definition for \@minipagerestore.)

\endminipage
460 \def\endminipage{%
461 \par
462 \unskip
463 \ifvoid\@mpfootins\else
464 \vskip\skip\@mpfootins
465 \normalcolor
466 \footnoterule
467 \unvbox\@mpfootins
468 \fi
469 \c@minipagefalse %% added 24 May 89
470 \color@endgroup
471 \egroup
472 \expandafter\c@iiparbox\ompargs{\unvbox\@tempboxa}

(End of definition for \endminipage.)

\@mplistdepth Versions of \c@listdepth and \footins local to minipage.
\@mpfootins 473 \newcount\@mplistdepth
474 \newinsert\@mpfootins

(End of definition for \@mplistdepth and \@mpfootins.)

\@mpfootnotetext Minipage version of \c@footnotetext.
Final \strut added 27 Mar 89, on suggestion by Don Hosek
475 </2ekernel>
476 <*2ekernel | latexrelease>
477 <latexrelease>\IncludeInRelease{2021/11/15}%
478 <latexrelease> {\@mpfootnotetext}{footnotetext tagging}%
479 \long\def\@mpfootnotetext#1{%
480 \global\setbox\@mpfootins\vbox{%
481 \unvbox\@mpfootins
482 \reset@font\footnotesize
483 \hsize\columnwidth
484 \parboxrestore
485 \def\@currentcounter{mpfootnote}%
486 \protected@edef\@currentlabel
487 {\csname p@mpfootnote\endcsname\@thefnmark}%
488 \color@begingroup
489 \makefntext{%
490 \rule\z@\footnotesep\ignorespaces#1\finalstrut\strutbox}%

```

```

491 \par
492 \color@endgroup}
493 </2ekernel | latexrelease>
494 \EndIncludeInRelease

495 \IncludeInRelease{2021/06/01}%
496 \long\def\@mpfootnotetext{\footnotetext tagging}%
497 \long\def\@mpfootnotetext#1{%
498 \global\setbox\@mpfootins\vbox{%
499 \unvbox\@mpfootins
500 \reset@font\footnotesize
501 \hsize\columnwidth
502 \parboxrestore
503 \protected@edef\@currentlabel
504 {\csname p@mpfootnote\endcsname\@thefnmark}%
505 \color@begingroup
506 \makefntext{%
507 \rule{z@footnotesep}{ignorespaces#1}\@finalstrut\strutbox}%
508 \par
509 \color@endgroup}
510 \EndIncludeInRelease

511 \IncludeInRelease{0000/00/00}%
512 \long\def\@mpfootnotetext{\footnotetext tagging}%
513 \long\def\@mpfootnotetext#1{%
514 \global\setbox\@mpfootins\vbox{%
515 \unvbox\@mpfootins
516 \reset@font\footnotesize
517 \hsize\columnwidth
518 \parboxrestore
519 \protected@edef\@currentlabel
520 {\csname p@mpfootnote\endcsname\@thefnmark}%
521 \color@begingroup
522 \makefntext{%
523 \rule{z@footnotesep}{ignorespaces#1}\@finalstrut\strutbox}%
524 \color@endgroup}
525 \EndIncludeInRelease
526 *2ekernel}
527
```

*(End of definition for \@mpfootnotetext.)*

```
529 \newif\if@pboxsw
```

**\rule** Draw a rule of the specified size.

```

530 </2ekernel>
531 \IncludeInRelease{2015/01/01}%
532 \long\def\@rule{\Make\@rule robust}%
533 *2ekernel | latexrelease>
534 \DeclareRobustCommand\@rule{\@ifnextchar[\@rule[\@rule[\z@]]}%
535 </2ekernel | latexrelease>
536 \EndIncludeInRelease
537 \IncludeInRelease{0000/00/00}%
538 \long\def\@rule{\Make\@rule robust}%
539 \def\@rule{\@ifnextchar[\@rule[\@rule[\z@]]}%
```

```

540 ⟨latexrelease⟩\expandafter\let\csname rule \endcsname\@undefined
541 ⟨latexrelease⟩\EndIncludeInRelease
542 ⟨*2ekernel⟩

```

(*End of definition for \rule.*)

**\@rule** Internal form of \rule.

```

543 \def\@rule[#1]#2#3{%
544 \leavevmode
545 \hbox{%
546 \setlength\@tempdima{#1}%
547 \setlength\@tempdimb{#2}%
548 \setlength\@tempdimc{#3}%
549 \advance\@tempdimc\@tempdima
550 \vrule@width\@tempdimb@height\@tempdimc@depth-\@tempdima}}

```

(*End of definition for \@rule.*)

**\@@underline** Saved primitive \underline.

```
551 \let\@@underline\underline
```

(*End of definition for \@@underline.*)

**\underline** L<sup>A</sup>T<sub>E</sub>X version works outside math.

```

552 \DeclareRobustCommand\underline[1]{%
553 \relax
554 \ifmmode\@@underline{#1}%
555 \else $@\@@underline{\hbox{#1}}\m@th$\relax\fi}

```

(*End of definition for \underline.*)

**\raisebox** Raise a box, and change its vertical dimensions.

```

556 ⟨/2ekernel⟩
557 ⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
558 ⟨latexrelease⟩ {\raisebox}{\Make \raisebox robust}%
559 ⟨*2ekernel | latexrelease⟩
560 \DeclareRobustCommand\raisebox[1]{%
561 \leavevmode
562 \@ifnextchar[\{\@rsbox{#1}\}{\@irsbox{#1}[]}]
563 ⟨/2ekernel | latexrelease⟩
564 ⟨latexrelease⟩\EndIncludeInRelease
565 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
566 ⟨latexrelease⟩ {\raisebox}{\Make \raisebox robust}%
567 ⟨latexrelease⟩\def\raisebox#1{%
568 ⟨latexrelease⟩ \leavevmode
569 ⟨latexrelease⟩ \ifnextchar[\{\@rsbox{#1}\}{\@irsbox{#1}[]}]
570 ⟨latexrelease⟩\expandafter\let\csname raisebox \endcsname\@undefined
571 ⟨latexrelease⟩\EndIncludeInRelease
572 ⟨*2ekernel⟩

```

(*End of definition for \raisebox.*)

**\@rsbox** Optional argument handling.

```

573 \def\@rsbox#1[#2]{%
574 \ifnextchar[\{\@irsbox{#1}[#2]\}{\@irsbox{#1}[#2]}}

```

(End of definition for \@rsbox.)

\@argsbox ...

(End of definition for \@argsbox.)

\@irsbox Internal version of \raisebox (less than two optional args).

```
575 \long\def\@irsbox#1[#2]#3{%
576 \begin{tempboxa}\hbox{#3}%
577 \setlength{\tempdima{#1}}%
578 \ifx\#2\\\else\setlength{\tempdimb{#2}}\fi
579 \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
580 \ifx\#2\\\else\ht\@tempboxa\@tempdimb\fi
581 \box\@tempboxa
582 } \end{tempboxa}
```

(End of definition for \@irsbox.)

\@iirsbox Internal version of \raisebox (two optional args).

```
583 \long\def\@iirsbox#1[#2][#3]{%
584 \begin{tempboxa}\hbox{#4}%
585 \setlength{\tempdima{#1}}%
586 \setlength{\tempdimb{#2}}%
587 \setlength{\dimen@{#3}}%
588 \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
589 \ht\@tempboxa\@tempdimb
590 \dp\@tempboxa\dimen@
591 \box\@tempboxa
592 } \end{tempboxa}
```

(End of definition for \@iirsbox.)

\@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.

In 2024 we changed the macro to account for vertical mode. In that case we use a strut produced with \rule to avoid starting a new paragraph (resulting in spurious extra line) and also account for the \prevdepth of the previous line.

```
593 </2ekernel>
594 <*2ekernel | latexrelease>
595 <latexrelease>\IncludeInRelease{2024/06/01}%
596 <latexrelease> {\@finalstrut}{final strut correction}%
597 \def\@finalstrut#1{%
598 \unskip
599 \ifhmode \nobreak
600 \else
```

If we are in vmode we now back up by a baseline.

```
601 \vskip-\baselineskip
602 \fi
```

Finally we unconditionally use `\vrule`.

```
603 \vrule\@width\z@\@height\z@\@depth\dp#1}
604 </2ekernel | latexrelease>
605 <latexrelease>\EndIncludeInRelease
606 <latexrelease>\IncludeInRelease{0000/00/00}%
607 <latexrelease> {\@finalstrut}{final strut correction} %
608 <latexrelease>\def\@finalstrut#1{ %
609 <latexrelease> \unskip\ifhmode\nobreak\fi
610 <latexrelease> \vrule\@width\z@\@height\z@\@depth\dp#1}
611 <latexrelease>
612 <latexrelease>\EndIncludeInRelease
613 <*2ekernel>
```

(*End of definition for `\@finalstrut`.*)

## 1.1 Some low-level constructs

The following commands are basically inherited from plain TeX.

`\leftline` These macros place text on a full line either centred or left or right adjusted.  
`\rightline` 614 `\def\@cline{\hb@xt@{\hsize}`  
`\centerline` 615 `\ DeclareRobustCommand\leftline[1]{\@cline{\#1\hss}}`  
`\@cline` 616 `\ DeclareRobustCommand\rightline[1]{\@cline{\hss\#1}}`  
617 `\ DeclareRobustCommand\centerline[1]{\@cline{\hss\#1\hss}}`

(*End of definition for `\leftline` and others.*)

`\rlap` These macros place text to the left or right of the current reference point without taking  
`\llap` up space.  
`\clap` 618 `\ DeclareRobustCommand\rlap[1]{\hb@xt@{\z@{\#1\hss}}`  
619 `\ DeclareRobustCommand\llap[1]{\hb@xt@{\z@{\hss\#1}}`

And here is the version that centers, it was initially introduced by `mathtools`.

```
620 \DeclareRobustCommand\clap[1]{\hb@xt@{\z@{\hss\#1\hss}}}
```

(*End of definition for `\rlap`, `\llap`, and `\clap`.*)

```
621 </2ekernel>
```

## File 41

# 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 L<sup>A</sup>T<sub>E</sub>X 2.09 version, not the extended version described in *The L<sup>A</sup>T<sub>E</sub>X Companion*. Use the `array` package to obtain the extended version.

#### 1.1 tabbing

*Historical L<sup>A</sup>T<sub>E</sub>X 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 \totallmargin, 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 := \totallmargin + \linewidth
 \hbox{\@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
 \@heighttab := \@nxttabmar :=G \@firsttab
 \@tabpush :=G 0
 \dimen\@firsttab := \@totallleftmargin
 @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 < \@heighttab
 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 L<sup>A</sup>T<sub>E</sub>X 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 of definition for \a.)*

The ‘2ekernel’ code ensures that a \usepackage{autotabg} is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

1 \expandafter\let\csname ver@autotabg.sty\endcsname\fmtversion

```

\@firsttab
\@maxtab
 1 \expandafter\let\csname ver@autotabg.sty\endcsname\fmtversion
 2 \newdimen\@tempa
 3 \chardef\@firsttab=\the\allocationnumber
 4 \newdimen\@tempa\newdimen\@tempa\newdimen\@tempa\newdimen\@tempa
 5 \newdimen\@tempa\newdimen\@tempa\newdimen\@tempa\newdimen\@tempa
 6 \newdimen\@tempa\newdimen\@tempa\newdimen\@tempa\newdimen\@tempa
 7 \newdimen\@tempa\newdimen\@tempa\newdimen\@tempa\newdimen\@tempa
 8 \newdimen\@tempa
 9 \chardef\@maxtab=\the\allocationnumber
10 \dimen\@firsttab=0pt

```

(End of definition for `\@firsttab` and `\@maxtab`.)

```
\@nxttabmar
\@curtabmar 11 \newcount\@nxttabmar
\@curtab 12 \newcount\@curtabmar
\@heightab 13 \newcount\@curtab
\@tabpush 14 \newcount\@heightab
15 \newcount\@tabpush
```

(End of definition for `\@nxttabmar` and others.)

```
\@curline
\@curfield 16 \newbox\@curline
\@tabbbox 17 \newbox\@curfield
18 \newbox\@tabbbox
```

(End of definition for `\@curline`, `\@curfield`, and `\@tabbbox`.)

```
\if@rjfield
19 \newif\if@rjfield
```

(End of definition for `\if@rjfield`.)

`\@startline` It is, in some sense, an error if the current margin tab setting is higher than the value of `\@heightab` (which is a local variable). That this is allowed is a fundamental design flaw which is not going to be corrected now.

```
20 \def\@startline{
21 \ifnum \@nxttabmar >\@heightab
22 \@badtab
23 \global\@nxttabmar \@heightab
24 \fi
25 \global\@curtabmar \@nxttabmar
26 \global\@curtab \@curtabmar
27 \global\setbox\@curline \hbox {}%
28 \@startfield
29 \strut}
```

(End of definition for `\@startline`.)

```
\@stopline
30 \def\@stopline{
31 \unskip
32 \@stopfield
33 \if@rjfield
34 \global\@rjfieldfalse
35 \@tempdima\@totalleftmargin
36 \advance\@tempdima\linewidth
37 \hb@xt@\@tempdima{
38 \itemfudge\hskip\dimen\@curtabmar
39 \box\@curline
40 \hfil
41 \box\@curfield} %
42 \else
43 \addfield
44 \hbox{\itemfudge\hskip\dimen\@curtabmar\box\@curline} %
45 \fi}
```

(End of definition for \@stopline.)

\@startfield

```
46 \def\@startfield{%
47 \global\setbox\@curfield\hbox\bgroup\color@begingroup}
```

(End of definition for \@startfield.)

\@stopfield

```
48 \def\@stopfield{%
49 \color@endgroup\egroup}
```

(End of definition for \@stopfield.)

\@contfield

```
50 \def\@contfield{%
51 \global\setbox\@curfield\hbox\bgroup\color@begingroup
52 \unhbox\@curfield}
```

(End of definition for \@contfield.)

\@addfield

```
53 \def\@addfield{\global\setbox\@curline\hbox{\unhbox
54 \@curline\unhbox\@curfield}}
```

(End of definition for \@addfield.)

\@ifatmargin

```
55 \def\@ifatmargin{\ifdim \wd\@curline =\z@}
```

(End of definition for \@ifatmargin.)

\@tabcr

```
56 \protected\def\@tabcr{\@stopline \@ifstar{\penalty \OM \@xtabcr}\@xtabcr}
```

(End of definition for \@tabcr.)

\@xtabcr

```
57 \def\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}
```

(End of definition for \@xtabcr.)

\@itabcr

```
58 </2ekernel>
59 <*2ekernel | latexrelease>
60 <latexrelease>\IncludeInRelease{2020/10/01}%
61 <latexrelease> {\@itabcr}{Tabbing calc syntax}%
62 \def\@itabcr[#1]{\@vspace@calcify{#1}\@startline\ignorespaces}
63 </2ekernel | latexrelease>
64 <latexrelease>\EndIncludeInRelease
65 <latexrelease>\IncludeInRelease{0000/00/00}%
66 <latexrelease> {\@itabcr}{Tabbing calc syntax}%
67 <latexrelease>
68 <latexrelease>\def\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
69 <latexrelease>\EndIncludeInRelease
70 <*2ekernel>
```

```

tabbing (env.) We use \relax to prevent \item from scanning too far.

\tabbing 71 \def\tabbing{\lineskip \z@skip\let\>\@rtab\let\<\@ltab\let\=\@settab
72 \let\+\@tabplus\let\-\@tabminus\let\`{@tabrj\let\`\@tablab
73 \let\\=\@tabcr
74 \chightab\firstab
75 \global\@nxttabmar\firstab
76 \dimen\firstab\@totalleftmargin
77 \global\@tabpush\z@\global\@rjfieldfalse
78 \trivlist \item\relax
79 \if@minipage\else\vskip\parskip\fi
80 \setbox\@tabfbox\hbox{%
81 \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
82 \def\@itemfudge{\box\@tabfbox}%
83 \startline\ignorespaces}

\endtabbing 84 \def\endtabbing{%
85 \stopline\ifnum\@tabpush >\z@\badpoptabs \fi\endtrivlist}

Omitted \global added to \@rtab 17 Jun 86

\@rtab 86 \def\@rtab{\@stopfield\@addfield\ifnum \@curtab<\chightab
87 \global\advance\@curtab \one\else\badtab\fi
88 \tempdima\dimen\@curtab
89 \advance\@tempdima -\dimen\@curtabmar
90 \advance\@tempdima -\wd\curline
91 \global\setbox\curline\hbox{\unhbox\curline\hskip\@tempdima}%
92 \startfield\ignorespaces}

\@settab 93 \def\@settab{\@stopfield\@addfield
94 \ifnum \@curtab <\maxtab
95 \ifnum\@curtab =\chightab
96 \advance\chightab \one
97 \fi
98 \global\advance\@curtab \one
99 \else
100 \@latex@error{Tab overflow}\ehd
101 \fi
102 \dimen\@curtab \dimen\@curtabmar
103 \advance\dimen\@curtab \wd\curline
104 \startfield
105 \ignorespaces}

\@ltab 106 \def\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firstab
107 \global\advance\@curtab \m@ne \global\advance\@curtabmar \m@ne\else
108 \badtab\fi\else
109 \@latex@error{\string\<\space in mid line}\ehd\fi\ignorespaces}

\@tabplus 110 \def\@tabplus{%
111 \ifnum\@nxttabmar<\chightab

```

```

112 \global\advance\@nxttabmar\@ne
113 \else
114 \@badtab
115 \fi
116 \ignorespaces}

\@tabminus 117 \def\@tabminus{%
118 \ifnum\@nxttabmar>\@firsttab
119 \global\advance\@nxttabmar\m@ne
120 \else
121 \@badtab
122 \fi
123 \ignorespaces}

\@tabrj 124 \def\@tabrj{%
125 \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}

\setbox\@curline made \global in \@tablab. 17 Jun 86

\@tablab 126 \def\@tablab{%
127 \@stopfield
128 \global\setbox\@curline\hbox{%
129 \box\@curline
130 \hskip-\wd\@curfield \hskip-\tabbingsep
131 \box\@curfield
132 \hskip\tabbingsep}%
133 \@startfield
134 \ignorespaces}

135 </2ekernel>
136 <*2ekernel | latexrelease>
137 <latexrelease>\IncludeInRelease{2019/10/01}%
138 <latexrelease> {\pushtabs}{Make commands robust}%

\pushtabs 139 \DeclareRobustCommand\pushtabs{%
140 \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
141 \@contfield}

```

It is, in some sense, an error if, after the endgroup, the current tab setting is higher than the new value of \chightab (which is a local variable). That this is allowed is a fundamental design flaw which is not going to be corrected now.

```

142 \DeclareRobustCommand\poptabs{\@stopfield\@addfield
143 \ifnum \@tabpush >\z@
144 \endgroup
145 \global\advance\@tabpush \m@ne
146 \ifnum \@curtab >\chightab
147 \global \@curtab \chightab
148 \@badtab
149 \fi
150 \else
151 \@badpoptabs
152 \fi
153 \@contfield}

```

```

154 \DeclareRobustCommand\kill{\@stopfield\@startline\ignorespaces}

(End of definition for \itabcr and others.)

155 </2ekernel | latexrelease>
156 <latexrelease>\EndIncludeInRelease
157 <latexrelease>\IncludeInRelease{0000/00/00}%
158 <latexrelease> {\pushtabs}{Make commands robust}%
159 <latexrelease>
160 <latexrelease>\kernel@make@fragile\pushtabs
161 <latexrelease>\kernel@make@fragile\poptabs
162 <latexrelease>\kernel@make@fragile\kill
163 <latexrelease>
164 <latexrelease>\EndIncludeInRelease
165 <*2ekernel>

\tabbingsep
166 \newdimen\tabbingsep

(End of definition for \tabbingsep.)

```

## 1.2 array and tabular environments

*Historical L<sup>A</sup>T<sub>E</sub>X 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<br>placing a strut in every row of height and depth<br>\arraystretch times the height and depth of the strut<br>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.<br>\arraycolsep or \tabcolsep spacing is suppressed.         |
| *{N}{PRE} | : equivalent to writing N copies of PRE in the preamble.<br>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
 \leavemode
 \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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

- \extracolsep This command needs to expand during the tabular preamble construction so can't be robust.

167 \def\extracolsep#1{\tabskip #1\relax}

*(End of definition for \extracolsep.)*

\array(*array*)

168 \def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz  
169 \let\@classiv\@arrayclassiv  
170 \let\\@\@arraycr\let\@halignto\@empty\@tabarray}

*(End of definition for \array.)*

\endarray

\endtabular 171 \def\endarray{\crcr\egroup\egroup}  
\endtabular\* 172 \def\endtabular{\crcr\egroup\egroup \$}\egroup  
173 \expandafter \let \csname endtabular\*\endcsname = \endtabular

*(End of definition for \endarray, \endtabular, and \endtabular\*.)*

\tabular(*tabular*)

174 \def\tabular{\let\@halignto\@empty\@tabular}

*(End of definition for \tabular.)*

\tabular\*

- Note that the change to use \setlength slightly alters the timing of the expansion and use of the length in #1 but this is very unlikely to have any practical effect.

175 \namedef{tabular\*}{#1}{%  
176 \setlength\dimen@{#1}{%  
177 \edef\@halignto{to\the\dimen@}\@tabular}

*(End of definition for \tabular\*.)*

\@tabular

178 \def\@tabular{\leavevmode \hbox \bgroup \$\let\@acol\@tabacol  
179 \let\@classz\@tabclassz  
180 \let\@classiv\@tabclassiv \let\\@\@tabularcr\@tabarray}

*(End of definition for \@tabular.)*

\@tabarray RmS 91/11/04 added \m@th.

181 \def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}

*(End of definition for \@tabarray.)*

RmS 1993/11/03 changed \halign to \ialign and removed superfluous \tabskip assignment

\@array

182 \def\@array[#1]{#2}{%  
183 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi

```
184 \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.

```
185 \setbox\@arstrutbox\hbox{%
186 \vrule \height\arraystretch\ht\strutbox
187 \depth\arraystretch \dp\strutbox
188 \width\z@\%
189 \mkpream{#2}%
190 \edef\@preamble{%
191 \ialign \noexpand\@halignto
192 \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
193 }
```

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.

```
193 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
194 \let\tabularnewline\\%
195 \let\par\empty
196 \let\sharp##%
197 \set@typeset@protect
198 \lineskip\z@skip\baselineskip\z@skip
```

If the parsing of the preamble goes wrong there may be some characters left which TeX then tries to typeset, i.e., we would be in horizontal mode. That would produce an endless loop because the `\halign` expects vertical mode thus issues a `\par` but that is a no-op at this point. So we better test this case issue some error message and make a crude recovery by ending that horizontal mode with force. A better fix would be to ensure that we never pick up more than a single character token (not done).

```
199 \ifhmode \preamerr\z@ \par\fi
200 \@preamble}
```

(End of definition for `\@array`.)

`\@arraycr` Array version of `\``.

```
201 \protected\def\@arraycr{%
202 ${\ifnum0='}\fi\@ifstar\@xarraycr\@xarraycr}
```

(End of definition for `\@arraycr`.)

`\@arraycr`

```
203 \def\@xarraycr{\@ifnextchar[\@garraycr{\ifnum0='{\fi}{$}\cr}}
```

(End of definition for `\@arraycr`.)

`\@garraycr`

```
204 \def\@garraycr[#1]{%
205 \ifnum0='{\fi}{$}\ifdim #1>\z@ \@xarraycr[#1]\else
206 \@yarraycr[#1]\fi}
```

(End of definition for `\@garraycr`.)

```

\tabularnewline Tabular version of \\.
207 \let\tabularnewline\relax
(End of definition for \tabularnewline.)
```

```

\@tabularcr
208 \protected\def\@tabularcr{%
209 {\ifnum0='}\fi\@ifstar\@xtabularcr\@xtabularcr}
(End of definition for \@tabularcr.)
```

```

\@xtabularcr
210 \def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0='{\fi}\cr}}
(End of definition for \@xtabularcr.)
```

```

\@argtabularcr
211 \def\@argtabularcr[#1]{%
212 \ifnum0='{\fi}%
213 \ifdim #1>\z@%
214 \unskip\@xargarraycr{#1}%
215 \else%
216 \@yargarraycr{#1}%
217 \fi}
(End of definition for \@argtabularcr.)
```

```

\@xargarraycr
218 \def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \arstrutbox
219 \vrule \height\z@ \depth\@tempdima \width\z@ \cr}
(End of definition for \@xargarraycr.)
```

```

\@yargarraycr
220 </2ekernel>
221 <*2ekernel | latexrelease>
222 <latexrelease>\IncludeInRelease{2020/10/01}%
223 <latexrelease> {\@yargarraycr}{tabular support calc syntax}%
224 \def\@yargarraycr#1{\cr\noalign{\vspace@calcify{#1}}}
225 </2ekernel | latexrelease>
226 <latexrelease>\EndIncludeInRelease
227 <latexrelease>\IncludeInRelease{0000/00/00}%
228 <latexrelease> {\@yargarraycr}{tabular support calc syntax}%
229 <latexrelease>
230 <latexrelease>\def\@yargarraycr#1{\cr\noalign{\vskip #1}}
231 <latexrelease>\EndIncludeInRelease
232 <*2ekernel>
(End of definition for \@yargarraycr.)
```

\multicolumn *Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
\multicolumn{NUMBER}{FORMAT}{ITEM} ==
BEGIN
\multispan{NUMBER}
\begingroup
\caddamp == null
\mkpream{FORMAT}
\sharp == ITEM
\protect == \relax
\startpbox == \@@startpbox
\endpbox == \@@endpbox
\carstrut
\preamble
\endgroup
END
```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

The command \def\caddamp{} 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.

```
233 \long\def\multicolumn#1#2#3{\multispan{#1}\begingroup
234 \mkpream{#2}%
235 \def\sharp{#3}\set@typeset@protect
236 \let\startpbox\@@startpbox\let\endpbox\@@endpbox
237 \carstrut \preamble\hbox{}\endgroup\ignorespaces}
```

*(End of definition for \multicolumn.)*

*Historical L<sup>A</sup>T<sub>E</sub>X 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 @firststamp = true then @firststamp := false
else & fi
END

\@mkpream TOKENLIST ==
BEGIN
 @firststamp := 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$\relax\sharp$\hfil
 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\@c END

\@xexpast S1 *{N}{S2} S3 \@c ==
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 \@c
END
End of historical LATEX 2.09 comments.
```

```
\@xexnoop
238 \def\@xexnoop #1\@c{}

(End of definition for \@xexnoop.)

\@expast
239 \def\@expast#1{\@xexpast #1*0x\@c}

(End of definition for \@expast.)
```

```

\@xexpast

240 \def\@xexpast#1##2##3##4@@{%
241 \edef\reserved@a{#1}%
242 \tempcnta#2\relax
243 \ifnum\tempcnta>\z@
244 \whilenum\tempcnta>\z@\do
245 {\edef\reserved@a{\reserved@a#3}\advance\tempcnta \m@ne}%
246 \let\reserved@b\@xexpast
247 \else
248 \let\reserved@b\@xexnoop
249 \fi
250 \expandafter\reserved@b\reserved@a #4@@}

```

(End of definition for `\@xexpast`.)

```

\if@firstamp
\@addamp 251 \newif\if@firstamp
252 \def\@addamp{%
253 \if@firstamp
254 \if@firstampfalse
255 \else
256 \edef\@preamble{\@preamble &}%
257 \fi}

```

(End of definition for `\if@firstamp` and `\@addamp`.)

```

\@arrayacol
\@tabacol 258 \def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\@ampacol 259 \def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\@cacolampacol 260 \def\@ampacol{\@addamp \@acol}
261 \def\@cacolampacol{\@acol\@addamp\@acol}

```

(End of definition for `\@arrayacol` and others.)

```

\@mkpream
262 \def\@mkpream#1{\@firstamptrue\@lastchclass6
263 \let\@preamble\empty
264 \let\protect\unexpandable\protect
265 \let\@sharp\relax
266 \let\@startpbox\relax\let\@endpbox\relax
267 \expandafter\@tfor \expandafter
268 \nextchar \expandafter:\expandafter=\reserved@a\do
269 {\@testpach\@nextchar
270 \ifcase \chclass \classz \or \classi \or \classii \or \classiii
271 \or \classiv \or \classv \fi\@lastchclass\chclass}%
272 \ifcase \lastchclass \@acol
273 \or \or \preamerr \one\or \preamerr \tw@ \or \or \@acol \fi}

```

(End of definition for `\@mkpream`.)

```

\@arrayclassz

275 \def\@arrayclassz{\ifcase \lastchclass \acolampacol \or \campacol \or
276 \or \or \addamp \or
277 \acolampacol \or \firststampfalse \acol \fi
278 \edef\@preamble{\@preamble
279 \ifcase \chnum
280 \hfil$\relax\sharp\$hfil \or \$\relax\sharp\$hfil
281 \or \hfil$\relax\sharp\$fi\}}

```

(End of definition for \@arrayclassz.)

\@tabclassz RmS 91/08/14 inserted extra braces around entry for NFSS

```

282 \def\@tabclassz{%
283 \ifcase\lastchclass
284 \acolampacol
285 \or
286 \campacol
287 \or
288 \or
289 \or
290 \addamp
291 \or
292 \acolampacol
293 \or
294 \firststampfalse\acol
295 \fi
296 \edef\@preamble{%
297 \@preamble{%
298 \ifcase\chnum
299 \hfil
300 \hskip1sp%
301 \ignorespaces\sharp\unskip\hfil
302 \or
303 \hskip1sp\ignorespaces\sharp\unskip\hfil
304 \or
305 \hfil\hskip1sp\ignorespaces\sharp\unskip
306 \fi}}}

```

(End of definition for \@tabclassz.)

```

\@classi

307 \def\@classi{%
308 \ifcase\lastchclass
309 \acol\arrayrule
310 \or
311 \addtopreamble{\hskip \doublerulesep}\arrayrule
312 \or
313 \or
314 \or
315 \arrayrule
316 \or
317 \acol\arrayrule
318 \or

```

```

319 \@arrayrule
320 \fi}

(End of definition for \@classi.)
```

\@classii

```

321 \def\@classii{%
322 \ifcase\@lastchclass
323 \or
324 \addtopreamble{\hspace{.5\arrayrulewidth}%
325 \fi}

(End of definition for \@classii.)
```

\@classiii

```

326 \def\@classiii{\ifcase \@lastchclass \acolampacol \or
327 \addamp\acol \or
328 \or \or \addamp \or
329 \acolampacol \or \ampacol \fi}

(End of definition for \@classiii.)
```

\@tabclassiv

```

330 \def\@tabclassiv{\addtopreamble\@nextchar}

(End of definition for \@tabclassiv.)
```

\@arrayclassiv

```

331 \def\@arrayclassiv{\addtopreamble{$\@nextchar$} }
```

(End of definition for \@arrayclassiv.)

\@classv

```

332 \def\@classv{\addtopreamble{\startpbox{\@nextchar}\ignorespaces
333 \sharp\endpbox}}
```

(End of definition for \@classv.)

\@addtopreamble

```

334 \def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}
```

(End of definition for \@addtopreamble.)

\@chclass

\@lastchclass

```

335 \newcount\@chclass
336 \newcount\@lastchclass
337 \newcount\@chnum
```

(End of definition for \@chclass, \@lastchclass, and \@chnum.)

\arraycolsep

\tabcolsep

```

338 \newdimen\arraycolsep
339 \newdimen\tabcolsep
```

\arrayrulewidth

```

340 \newdimen\arrayrulewidth
341 \newdimen\doublerulesep
```

(End of definition for \arraycolsep and others.)

```
\arraystretch
342 \def\arraystretch{1} % Default value.
```

(End of definition for \arraystretch.)

```
\@arstrutbox
 \@arstrut 343 \newbox\@arstrutbox
 344 \def\@arstrut{
 345 \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}
```

(End of definition for \@arstrutbox and \@arstrut.)

```
\@arrayrule
346 \def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
347 \vrule \width \arrayrulewidth\hskip -.5\arrayrulewidth}}
```

(End of definition for \@arrayrule.)

```
\@testpach
348 \def\@testpach#1{\@chclass \ifnum \lastchclass=\tw@ 4 \else
349 \ifnum \lastchclass=3 5 \else
350 \z@ \if #1c\@chnum \z@ \else
351 \if #11\@chnum \one \else
352 \if #1r\@chnum \tw@ \else
353 \@chclass \if #1|\one \else
354 \if #1@\tw@ \else
355 \if #1p3 \else \z@ \@preamerr 0\fi
356 \fi \fi \fi \fi \fi
357 }
```

(End of definition for \@testpach.)

```
\hline
358 \def\hline{
359 \noalign{\ifnum0='}\fi\hrule \height \arrayrulewidth \futurelet
360 \reserved@a\@xhline}
```

(End of definition for \hline.)

```
\@xhline
361 \def\@xhline{\ifx\reserved@a\hline
362 \vskip\doublerulesep
Measure from the middle of the rules.
363 \vskip-\arrayrulewidth
364 \fi
365 \ifnum0='{\fi}}
```

(End of definition for \@xhline.)

```
\vline
366 \def\vline{\vrule \width \arrayrulewidth}
```

(End of definition for \vline.)

`\cline` The old L<sup>A</sup>T<sub>E</sub>X2.09 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 `lplain.dtx`.

These counters are no longer declared.

```

\newcount\@cla
\newcount\@clb

367 \def\cline#1{\@cline#1\@nil}

368 \def\@cline#1-#2\@nil{%
369 \omit

```

Use the counter from `\multispan`.

```

370 \multicnt#1%
371 \advance\multispan\m@ne
372 \ifnum\multicnt=\@ne\@firstofone{\&\omit}\fi
373 \multicnt#2%
374 \advance\multicnt-#1%
375 \advance\multispan\@ne

```

The original had `\unskip` at this point, but how could a skip get here ???

```

376 \leaders\hrule\@height\arrayrulewidth\hfill
377 \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.

```
378 \noalign{\vskip-\arrayrulewidth}}
```

(End of definition for `\cline` and `\@cline`.)

`\mscount` The `\mscount` counter is no longer declared, saving a csname and a register. It is declared in compatibility mode.

(End of definition for `\mscount`.)

`\multispan` Modify `\multispan` slightly from its plain T<sub>E</sub>X definition to allow more efficient code sharing with `\multicolumn`. Also share a count register with `\multiput`.

```

\sp@n 379 \def\multispan{\omit\@multispan}
380 \def\@multispan#1{%
381 \multicnt#1\relax
382 \loop\ifnum\multicnt>\@ne \sp@n\repeat}
383 \def\sp@n{\span\omit\advance\multicnt\m@ne}

```

(End of definition for `\multispan`, `\@multispan`, and `\sp@n`.)

`\@startpbox` Helper macros for ‘p’ columns.

```

\@endpbox 384 \def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
 \@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)

```

```
384 \def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
```

```
385 \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 of definition for \@startpbox and \@endpbox.)

```
\@@startpbox
\@@endpbox
386 \let\@@startpbox=\@startpbox
387 \let\@@endpbox=\@endpbox
```

(End of definition for \@startpbox and \@endpbox.)

388 ⟨/2ekernel⟩

## File 42

# ltpictur.dtx

### 1 Picture Mode

Picture mode commands. In addition to the commands available in L<sup>A</sup>T<sub>E</sub>X2.09, This section adds the new \qbezier command for drawing curves.

\qbezier \qbezier[ $\langle N \rangle$ ] ( $\langle AX,AY \rangle$ ) ( $\langle BX,BY \rangle$ ) ( $\langle CX,CY \rangle$ ) plots a quadratic Bezier curve from ( $\langle AX,AY \rangle$ ) to ( $\langle CX,CY \rangle$ ), with ( $\langle BX,BY \rangle$ ) as the third Bezier point, using  $N+1$  points equally spaced parametrically. If  $N = 0$  (the default value), then a sufficient number of points are used to draw a connected curve—except that at most \qbeziermax+1 points are drawn. A “point” is a square of side \@wholewidth.

\bezier In addition, to be compatible with the old **bezier** package, a variant of this command, \bezier, is defined, in which the first argument is not optional.

*Historical L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

The ‘2ekernel’ code ensures that a \usepackage{autopict} is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

1 <2ekernel>\expandafter\let\csname ver@autopict.sty\endcsname\fmtversion

```

\@wholewidth
\@halfwidth
 2 <*2ekernel>
 3 \newdimen\@wholewidth
 4 \newdimen\@halfwidth

```

*(End of definition for \@wholewidth and \@halfwidth.)*

```

\unitlength
 5 \newdimen\unitlength \unitlength =1pt

```

*(End of definition for \unitlength.)*

```

\@picbox
\@picht
 6 \newbox\@picbox
 7 \newdimen\@picht

```

*(End of definition for \@picbox and \@picht.)*

`\@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.

```
 8 </2ekernel>
 9 <*2ekernel | latexrelease>
10 <latexrelease>\IncludeInRelease{2020/10/01}%
11 <latexrelease> {\@defaultunitsset}{default units}%
12 \def\@defaultunitsset#1#2#3{%
13 \@defaultunits#1\dimexpr#2#3\relax\relax\@nnil}
14 </2ekernel | latexrelease>
```

This is used in all `picture` commands that take picture coordinates. So `\put(2,2)` as previously but now `\put(\textwidth-5cm,0.4\textheight)` Note that you can only use expressions with lengths, `\put(1+2,0)` is not supported.

```
15 <latexrelease>\EndIncludeInRelease
16 <latexrelease>\IncludeInRelease{0000/00/00}%
17 <latexrelease> {\@defaultunitsset}{default units}%
18 <latexrelease>\let\@defaultunitsset\@undefined
19 <latexrelease>\EndIncludeInRelease
20 <*2ekernel>
```

(End of definition for `\@defaultunitsset`.)

`pict\picture`) #1 should be white space.

#1 should be a ( (eating any white space before the bracket),

```
21 \long\def\picture#1{\pictur@#1}
22 \def\pictur@(#1){%
23 \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)})
```

(End of definition for `\picture` and `\pictur@`.)

`\@picture`

```
24 </2ekernel>
25 <*2ekernel | latexrelease>
26 <latexrelease>\IncludeInRelease{2020/10/01}%
27 <latexrelease> {\@picture}{default units}%
28 \def\@picture(#1,#2)(#3,#4){%
29 \@defaultunitsset\@picht{#2}\unitlength
30 \@defaultunitsset\@tempdimc{#1}\unitlength
31 \setbox\@picbox\hb@xt@\@tempdimc\bgroun
32 \@defaultunitsset\@tempdimc{#3}\unitlength
33 \hskip -\@tempdimc
34 \@defaultunitsset\@tempdimc{#4}\unitlength
35 \lower\@tempdimc\hbox\bgroun
36 \ignorespaces}
37 </2ekernel | latexrelease>
```

```

38 \end{macro}
39 \end{macro}
40 \def\@picture#1#2#3#4{%
41 \setbox\@picbox\hb@xt@#1\unitlength\bgroup
42 \picht#2\unitlength
43 \setbox\@picbox\hb@xt@#1\unitlength\bgroup
44 \hskip -#3\unitlength
45 \lower #4\unitlength\hbox\bgroup
46 \ignorespaces}
47 \end{macro}
48 \end{macro}

(End of definition for \@picture.)
```

\endpicture

```

49 \def\endpicture{%
50 \egroup\hss\egroup
51 \ht\@picbox\picht\dp\@picbox\z@%
52 \mbox{\box\@picbox}}
```

(End of definition for \endpicture.)

In the definitions of \put and \multiput, \hspace was replaced by \kern just in case arg #3 = “plus”. (Bug detected by Don Knuth. changed 20 Jul 87).

```

53 \end{macro}
54 \end{macro}
55 \end{macro}
56 \def\put#1#2#3{%
57 \expandafter\let\csname put \endcsname\@undefined
58 \long\def\put(#1,#2)#3{%
59 \killglue
60 \defaultunitsset\tempdimc{#2}\unitlength
61 \raise\tempdimc
62 \hb@xt@\z@{%
63 \defaultunitsset\tempdimc{#1}\unitlength
64 \kern\tempdimc
65 #3\hss}%
66 \ignorespaces}
67 \end{macro}
68 \end{macro}
69 \end{macro}
70 \def\multiput#1#2#3{%
71 \expandafter\let\csname put \endcsname\@undefined
72 \long\def\multiput(#1,#2)#3{%
73 \killglue\raise#2\unitlength
74 \hb@xt@\z@{\kern#1\unitlength #3\hss}%
75 \ignorespaces}
76 \end{macro}
77 \end{macro}

\multiput #3 had better be a .

78 \end{macro}
79 \end{macro}
80 \end{macro}
```

```

81 \begin{macro}{\multiput}
82 \expandafter\let\csname multiput \endcsname\undefined
83 \def\multiput(#1,#2){%
84 \ifdim#1=0pt \def\unitlength{\unitlength}%
85 \ifdim#2=0pt \def\unitlength{\unitlength}%
86 \multiput{}%
87 }%
```

(End of definition for `\multiput`.)

`\@multiput`

```

98 \begin{macro}{\@multiput}
99 \expandafter\let\csname @multiput \endcsname\undefined
100 \def\@multiput(#1,#2){%
101 \ifdim#1=0pt \def\unitlength{\unitlength}%
102 \ifdim#2=0pt \def\unitlength{\unitlength}%
103 \killglue\multicnt #3\relax
104 \whilenum \multicnt >\z@\do
105 {\raise\ydim\hb@xt@.z@{\kern\unitlength #4\hss}%
106 \advance\multicnt\m@ne
107 \ifdim#1>0pt \def\unitlength{\unitlength}%
108 \ifdim#2>0pt \def\unitlength{\unitlength}%
109 \ignorespaces}%
110 }%
```

(End of definition for `\@multiput`.)

`\@killglue`

```

123 \def\@killglue{\unskip\whiledim \lastskip >\z@\do{\unskip}}
```

(End of definition for `\@killglue`.)

```

\thinlines
\thicklines 124 \DeclareRobustCommand\thinlines{\let\@linefnt\tenln
125 \let\@circlefnt\tencirc
126 \wholewidth\fontdimen8\tenln \halfwidth .5\wholewidth}
127 \DeclareRobustCommand\thicklines{\let\@linefnt\tenlnw
128 \let\@circlefnt\tencircw
129 \wholewidth\fontdimen8\tenlnw \halfwidth .5\wholewidth}

(End of definition for \thinlines and \thicklines.)

\linethickness
130 \DeclareRobustCommand*\linethickness[1]
131 {\wholewidth #1\relax \halfwidth .5\wholewidth \ignorespaces}

(End of definition for \linethickness.)

\isshortstack
132 \def\shortstack{\ifnextchar[\@shortstack{\shortstack[c]}}
(End of definition for \isshortstack.)

\@isshortstack
133 \def\@shortstack[#1]{%
134 \leavevmode
135 \vbox\bgrou
136 \baselineskip-\p@\lineskip 3\p@
137 \let\mb@l\hss\let\mb@r\hss
138 \expandafter\let\csname mb@#1\endcsname\relax
139 \let\\@stackcr
140 \@isshortstack}

(End of definition for \@isshortstack.)

\@isshortstack
141 \def\@isshortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\crcr}\egroup}
(End of definition for \@isshortstack.)

\@stackcr
\@ixstackcr 142 \protected\def\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
143 \def\@ixstackcr{\ifnextchar[\@istackcr{\cr\ignorespaces}{}}

(End of definition for \@stackcr and \@ixstackcr.)

\@istackcr
144 </2ekernel>
145 <*2ekernel | latexrelease>
146 <latexrelease>\IncludeInRelease{2020/10/01}%
147 <latexrelease> {\@istackcr}{\shortstack calc support}%
148 \def\@istackcr[#1]{\cr\noalign{\@vspace@calcify{#1}}\ignorespaces}
149 </2ekernel | latexrelease>

```

```

150 \end{macro}
151 \end{macro}
152 \end{macro}
153 \end{macro}
154 \def\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}
155 \end{macro}
156 {*2ekernel}

(End of definition for \@istackcr.)
Historical LATEX 2.09 comments (not necessarily accurate any more):
\line(X,Y){LEN} ==
BEGIN
 \carg := X
 \yarg := Y
 \clinenlen := LEN * \unitlength
 if \carg = 0
 then \vline
 else if \yarg = 0
 then \hline
 else \sline
 if
 if
END

\sline ==
BEGIN
 if \carg < 0
 then @negarg := T
 \carg := -\carg
 \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\clinechar := \hbox{\clinefnt \getlinechar(\carg,\yyarg) }
 if \yarg > 0 then \upordown = \raise
 \clnht := 0
 else \upordown = \lower
 \clnht := height of \box\clinechar
 fi
 \clnwd := width of \box\clinechar
 if @negarg
 then \hskip - width of \box\clinechar
 \reserved@a == \hskip - 2* width of box \clinechar
 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
\@tempcpta := \@tempdima / width of \box\@linechar
\@tempdima := (\@tempcpta * 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
\@tempcpta := 8*X - 9
if Y > 0
 then \@tempcpta := \@tempcpta + Y
 else \@tempcpta := \@tempcpta - Y + 64
fi
\char\@tempcpta
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
 {\@clinefnt 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
 {\@clinefnt
 if @negarg then \@getlarrow(\@xarg,\@yyarg)
 else \@getrarrow(\@xarg,\@yyarg)
 fi }
 else error: 'LATEX ERROR: Illegal \line or \vector argument.'
 fi
END

\@getlarrow(X,Y) ==
BEGIN
 if Y = 0
 then \@tempcnta := '33
 else \@tempcnta := 16 * X - 9
 \@tempcntb := 2 * Y
 if \@tempcntb > 0
 then \@tempcnta := \@tempcnta + \@tempcntb
 else \@tempcnta := \@tempcnta - \@tempcntb + 64
 fi
 fi
 \char\@tempcnta
END

\@getrarrow(X,Y) ==
BEGIN

```

```

\@tempcntb := |Y|
case of \@tempcntb
 0 : \@tempcnta := '55
 1 : if X < 3
 then \@tempcnta := 24*X - 6
 else if X = 3
 then \@tempcnta := 49
 else \@tempcnta := 58 fi
 fi
 2 : if X < 3
 then \@tempcnta := 24*X - 3
 else \@tempcnta := 51 % X must = 3
 fi
 3 : \@tempcnta := 16*X - 2
 4 : \@tempcnta := 16*X + 7
endcase
if Y < 0
 then \@tempcnta := \@tempcnta + 64
fi
\char\@tempcnta
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

### \if@negarg

157 \newif\if@negarg

(*End of definition for \if@negarg.*)

### \line

```

158 〈/2ekernel〉
159 〈*2ekernel | latexrelease〉
160 〈| latexrelease〉\IncludeInRelease{2020/10/01}%
161 〈| latexrelease〉 {\line}{default units}%
162 〈| latexrelease〉\expandafter\let\csname line \endcsname\@undefined
163 \def\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
164 \@defaultunitsset\@linelen{#3}\unitlength
165 \ifdim\@linelen<\z@\@badlinearg\else
166 \ifnum\@xarg =\z@ \@vline
167 \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
168 \fi
169 \fi}
170 〈/2ekernel | latexrelease〉

171 〈| latexrelease〉\EndIncludeInRelease
172 〈| latexrelease〉\IncludeInRelease{0000/00/00}%
173 〈| latexrelease〉 {\line}{default units}%
174 〈| latexrelease〉\expandafter\let\csname line \endcsname\@undefined
175 〈| latexrelease〉\def\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
176 〈| latexrelease〉 \@linelen #3\unitlength
177 〈| latexrelease〉 \ifdim\@linelen<\z@\@badlinearg\else
178 〈| latexrelease〉 \ifnum\@xarg =\z@ \@vline
179 〈| latexrelease〉 \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
180 〈| latexrelease〉 \fi

```

```

181 〈latexrelease〉 \fi}
182 〈latexrelease〉\EndIncludeInRelease
183 〈*2ekernel〉

```

(End of definition for \line.)

\@sline

```

184 \def\@sline{%
185 \ifnum\@xarg<\z@ \negargtrue \xarg -\xarg \yyarg -\yarg
186 \else \negargfalse \yyarg \yarg \fi
187 \ifnum \yyarg >\z@ \tempcpta\yyarg \else \tempcpta -\yyarg \fi
188 \ifnum\tempcpta>6 \badlinearg\tempcpta\z@ \fi
189 \ifnum\@xarg>6 \badlinearg\@xarg \ne \fi
190 \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.

```

191 \ifdim\wd\@linechar=\z@
192 \setbox\@linechar\hbox{.}%
193 \badlinearg
194 \fi
195 \ifnum \yarg >\z@ \let\upordown\raise \clnht\z@
196 \else\let\upordown\lower \clnht \ht\@linechar\fi
197 \clnwd \wd\@linechar
198 \if@negarg
199 \hskip -\wd\@linechar \def\reserved@a{\hskip -2\wd\@linechar}%
200 \else
201 \let\reserved@a\relax
202 \fi
203 \whiledim \clnwd <\linelen \do
204 {\upordown\clnht\copy\@linechar
205 \reserved@a
206 \advance\clnht \ht\@linechar
207 \advance\clnwd \wd\@linechar}%
208 \advance\clnht -\ht\@linechar
209 \advance\clnwd -\wd\@linechar
210 \tempdima\linelen\advance\tempdima -\clnwd
211 \tempdimb\tempdima\advance\tempdimb -\wd\@linechar
212 \if@negarg \hskip -\tempdimb \else \hskip \tempdimb \fi
213 \multiply\tempdima \m
214 \tempcpta\tempdima
215 \tempdima \wd\@linechar \divide\tempcpta \tempdima
216 \tempdima \ht\@linechar \multiply\tempdima \tempcpta
217 \divide\tempdima \m
218 \advance\clnht \tempdima
219 \ifdim \linelen <\wd\@linechar
220 \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).

```

221 \ifdim \linelen = \z@
222 \else
223 \picture@warn
224 \fi

```

```

225 \else\@upordown\@clnht\copy\@linechar\fi}

(End of definition for \@sline.)
```

\@hline

```

226 \def\@hline{\ifnum \carg <\z@ \hskip -\@linelen \fi
227 \vrule \height \halfwidth \depth \halfwidth \width \@linelen
228 \ifnum \carg <\z@ \hskip -\@linelen \fi}
```

(End of definition for \@hline.)

\@getlinechar

```

229 \def\@getlinechar(#1,#2){\tempcnta#1\relax\multiply\tempcnta 8%
230 \advance\tempcnta -9\ifnum #2>\z@ \advance\tempcnta #2\relax\else
231 \advance\tempcnta -#2\relax\advance\tempcnta 64 \fi
232 \char\tempcnta}
```

(End of definition for \@getlinechar.)

\vector

```

233 </2ekernel>
234 (*2ekernel | latexrelease)
235 <latexrelease>\IncludeInRelease{2020/10/01}%
236 <latexrelease> {\vector}{default units}%
237 <latexrelease>\expandafter\let\csname vector \endcsname\undefined
238 \def\vector(#1,#2){\carg #1\relax \carg #2\relax
239 \tempcnta \ifnum\carg<\z@ -\carg\else\carg\fi
240 \ifnum\tempcnta<5\relax
241 \defaultunitsset\@linelen{#3}\unitlength
242 \ifdim\@linelen<\z@\badlinearg\else
243 \ifnum\carg =\z@ \vvector
244 \else \ifnum\carg =\z@ \hvector \else \svector\fi
245 \fi
246 \fi
247 \else\badlinearg\fi}
248 </2ekernel | latexrelease>

249 <latexrelease>\EndIncludeInRelease
250 <latexrelease>\IncludeInRelease{0000/00/00}%
251 <latexrelease> {\vector}{default units}%
252 <latexrelease>\expandafter\let\csname vector \endcsname\undefined
253 <latexrelease>\def\vector(#1,#2){\carg #1\relax \carg #2\relax
254 \tempcnta \ifnum\carg<\z@ -\carg\else\carg\fi
255 \ifnum\tempcnta<5\relax
256 \@linelen #3\unitlength
257 \ifdim\@linelen<\z@\badlinearg\else
258 \ifnum\carg =\z@ \vvector
259 \else \ifnum\carg =\z@ \hvector \else \svector\fi
260 \fi
261 \fi
262 \else\badlinearg\fi}
263 <latexrelease>\EndIncludeInRelease
264 (*2ekernel)
```

(End of definition for \vector.)

```

\@hvector
265 \def\@hvector{\@hline\hb@xt@\z@{\@linefnt
266 \ifnum \carg <\z@ \getlarrow(1,0)\hss\else
267 \hss\getrarrow(1,0)\fi}}
(End of definition for \@hvector.)
```

```

\@vvector
268 \def\@vvector{\ifnum \carg <\z@ \downvector \else \upvector \fi}
(End of definition for \@vvector.)
```

```

\@svector
269 \def\@svector{\@sline
270 \tempcnta\carg \ifnum\tempcnta <\z@ \tempcnta -\tempcnta\fi
271 \ifnum\tempcnta <5%
272 \hskip -\wd\@linechar
273 \upordown\clnht \hbox{\@linefnt \if@negarg
274 \getlarrow(\carg,\yyarg)\else \getrarrow(\carg,\yyarg)\fi}%
275 \else\badlinearg\fi}
(End of definition for \@svector.)
```

```

\@getlarrow
276 \def\@getlarrow(#1,#2){\ifnum #2=\z@ \tempcnta 27 \% '33
277 \else
278 \tempcnta #1\relax\multiply\tempcnta \sixt@n
279 \advance\tempcnta -9 \tempcntb #2\relax\multiply\tempcntb \tw@
280 \ifnum \tempcntb >\z@ \advance\tempcnta \tempcntb
281 \else\advance\tempcnta -\tempcntb\advance\tempcnta 64
282 \fi\fi\char\tempcnta}
(End of definition for \@getlarrow.)
```

```

\@getrarrow
283 \def\@getrarrow(#1,#2){\tempcntb #2\relax
284 \ifnum\tempcntb <\z@ \tempcntb -\tempcntb\relax\fi
285 \ifcase \tempcntb\relax \tempcnta 45 \% '55
286 \or
287 \ifnum #1<\thr@ \tempcnta #1\relax\multiply\tempcnta
288 24\advance\tempcnta -6 \else \ifnum #1=\thr@ \tempcnta 49
289 \else\tempcnta 58 \fi\fi\or
290 \ifnum #1<\thr@ \tempcnta=#1\relax\multiply\tempcnta
291 24\advance\tempcnta -\thr@ \else \tempcnta 51 \fi\or
292 \tempcnta #1\relax\multiply\tempcnta
293 \sixt@n \advance\tempcnta -\tw@ \else
294 \tempcnta #1\relax\multiply\tempcnta
295 \sixt@n \advance\tempcnta 7 \fi\ifnum #2<\z@ \advance\tempcnta 64 \fi
296 \char\tempcnta}
(End of definition for \@getrarrow.)
```

```

\@vline
297 \def\@vline{\ifnum \carg <\z@ \downline \else \upline\fi}
```

(End of definition for \@vline.)

```
\@upline
298 \def\@upline{%
299 \hb@xt@z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
300 \@height \@linelen \@depth \z@\hss}}

```

(End of definition for \@upline.)

```
\@downline
301 \def\@downline{%
302 \hb@xt@z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
303 \@height \z@ \@depth \@linelen \hss}}

```

(End of definition for \@downline.)

```
\@upvector
304 \def\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}%
305 \raise \@linelen \hb@xt@z@\lower \ht\@tempboxa\box\@tempboxa\hss}

```

(End of definition for \@upvector.)

```
\@downvector
306 \def\@downvector{\@downline\lower \@linelen
307 \hb@xt@z@\@linefnt\char 63 %
308 \hss]}

```

(End of definition for \@downvector.)

*Historical L<sup>A</sup>T<sub>E</sub>X 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
309 {/2ekernel}
310 {*2ekernel | latexrelease}
311 {latexrelease}\IncludeInRelease{2020/10/01}%
312 {latexrelease} {\dashbox}{default units}%
313 {latexrelease}\expandafter\let\csname dashbox \endcsname\@undefined
314 \def\dashbox#1(#2,#3){\leavevmode\hb@xt@z@{\baselineskip \z@skip
315 \lineskip \z@skip
316 \defaultunitsset\@dashdim{#2}\unitlength
317 \dashcnt \dashdim \advance\@dashcnt 200
318 \defaultunitsset\@dashdim{#1}\unitlength
319 \divide\@dashcnt \@dashdim
320 \ifodd\@dashcnt\@dashdim \z@
321 \advance\@dashcnt \one \divide\@dashcnt \tw@
322 \else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
323 \advance\@dashcnt \m@ne
324 \setbox\dashbox \hbox{\vrule \height \halfwidth \depth \halfwidth
325 \width \dashdim}\put(0,0){\copy\dashbox}%
326 \put(0,#3){\copy\dashbox}%
327 \put(#2,0){\hskip-\@dashdim\copy\dashbox}%
328 \put(#2,#3){\hskip-\@dashdim\box\dashbox}%
329 \multiply\@dashdim \thr@@
330 \fi
331 \setbox\dashbox \hbox{%
332 \defaultunitsset\@tempdimc{#1}\unitlength
333 \vrule \height \halfwidth \depth \halfwidth \width \@tempdimc
334 \hskip\@tempdimc}%
335 \tempcnta\z@
336 \put(0,0){\hskip\@dashdim \whilenum \tempcnta <\@dashcnt
337 \do{\copy\dashbox\advance\tempcnta \one }{\tempcnta\z@
338 \put(0,#3){\hskip\@dashdim \whilenum \tempcnta <\@dashcnt
339 \do{\copy\dashbox\advance\tempcnta \one }{%
340 \defaultunitsset\@dashdim{#3}\unitlength
341 \dashcnt \dashdim \advance\@dashcnt 200
342 \defaultunitsset\@dashdim{#1}\unitlength
343 \divide\@dashcnt \@dashdim
344 \ifodd\@dashcnt\@dashdim \z@
345 \advance\@dashcnt \one \divide\@dashcnt \tw@
346 \else

```

```

347 \divide\@dashdim \tw@ \divide\@dashcnt \tw@
348 \advance\@dashcnt \m@ne
349 \setbox\@dashbox\hbox{\hskip -\@halfwidth
350 \vrule \@width \@wholewidth
351 \@height \@dashdim}\put(0,0){\copy\@dashbox}%
352 \put(#2,0){\copy\@dashbox}%
353 \put(0,#3){\lower\@dashdim\copy\@dashbox}%
354 \put(#2,#3){\lower\@dashdim\copy\@dashbox}%
355 \multiply\@dashdim \thr@@
356 \fi
357 \@defaultunitsset\@tempdimb{#1}\unitlength
358 \setbox\@dashbox\hbox{%
359 \vrule \@width \@wholewidth \@height\@tempdimb}%
360 \z@\@tempcpta\z@
361 \put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcpta <\@dashcnt
362 \do{\vskip\@tempdimb\copy\@dashbox\advance\@tempcpta \@ne }%
363 \vskip\@dashdim}}\@tempcpta\z@
364 \put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcpta<\@dashcnt
365 \do{\vskip\@tempdimb\copy\@dashbox\advance\@tempcpta \@ne }%
366 \vskip\@dashdim}}\makepicbox(#2,#3)}
367
```

`</2ekernel | latexrelease>`

`<| latexrelease>\EndIncludeInRelease`

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

`<| latexrelease> \{ \dashbox \}{default units} %`

`<| latexrelease>\expandafter\let\csname dashbox \endcsname\@undefined`

`<| 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\@dashdim \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}\@tempcpta\z@`

`<| latexrelease>\put(0,0){\hskip\@dashdim \@whilenum \@tempcpta <\@dashcnt`

`<| latexrelease>\do{\copy\@dashbox\advance\@tempcpta \@ne } }\@tempcpta\z@`

`<| latexrelease>\put(0,#3){\hskip\@dashdim \@whilenum \@tempcpta <\@dashcnt`

`<| latexrelease>\do{\copy\@dashbox\advance\@tempcpta \@ne } }%`

`<| latexrelease>\@dashdim #3\unitlength`

`<| latexrelease>\@dashcnt \@dashdim \advance\@dashcnt 200`

`<| latexrelease>\@dashdim #1\unitlength\divide\@dashcnt \@dashdim`

`<| latexrelease>\ifodd\@dashcnt \@dashdim \z@`

```

401 〈\latexrelease〉\advance\@dashcnt \@ne \divide\@dashcnt \tw@
402 〈\latexrelease〉\else
403 〈\latexrelease〉\divide\@dashdim \tw@ \divide\@dashcnt \tw@
404 〈\latexrelease〉\advance\@dashcnt \m@ne
405 〈\latexrelease〉\setbox\@dashbox\hbox{\hskip -\@halfwidth
406 〈\latexrelease〉\vrule \@width \@wholewidth
407 〈\latexrelease〉\@height \@dashdim\put(0,0){\copy\@dashbox}%
408 〈\latexrelease〉\put(#2,0){\copy\@dashbox}%
409 〈\latexrelease〉\put(0,#3){\lower\@dashdim\copy\@dashbox}%
410 〈\latexrelease〉\put(#2,#3){\lower\@dashdim\copy\@dashbox}%
411 〈\latexrelease〉\multiply\@dashdim \thr@@
412 〈\latexrelease〉\fi
413 〈\latexrelease〉\setbox\@dashbox\hbox{\vrule \@width \@wholewidth
414 〈\latexrelease〉\@height #1\unitlength}\@tempcnta\z@
415 〈\latexrelease〉\put(0,0){%
416 〈\latexrelease〉 \hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
417 〈\latexrelease〉 \do{\vskip #1\unitlength\copy\@dashbox
418 〈\latexrelease〉 \advance\@tempcnta\@ne }%
419 〈\latexrelease〉 \vskip\@dashdim}\@tempcnta\z@
420 〈\latexrelease〉\put(#2,0){%
421 〈\latexrelease〉 \hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
422 〈\latexrelease〉 \do{\vskip #1\unitlength\copy\@dashbox
423 〈\latexrelease〉 \advance\@tempcnta \@ne }%
424 〈\latexrelease〉 \vskip\@dashdim}\}@\makepicbox(#2,#3)}
425 〈\latexrelease〉\EndIncludeInRelease
426 〈*2ekernel〉

```

(End of definition for \dashbox.)

*Historical L<sup>A</sup>T<sub>E</sub>X 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{
 \circleft
 \tempcpta := \tempcpta + 2
 \char \tempcpta
 \tempcpta := \tempcpta - 1
 \char \tempcpta
 \kern -2\tempdima
 \tempcpta := \tempcpta + 2
 \raise \tempdima \hbox { \char \tempcpta }
 \raise \tempdima \box\tempboxa
 }
 \ht\tempboxa := \dp\tempboxa := 0
 \put{-\ovro}{-\ovro}{\tempboxa}
 else
 \circ{\@tempdimb}{96}
 fi
\endgroup
END

\circle*{DIAM} == \dot{DIAM} == \circ{DIAM*\unitlength}{112}

\circ{DIAM}{CHAR} ==
BEGIN

```

```

\@tempcnta := integer coercion of (DIAM + .5pt)/1pt.
if \@tempcnta > 15 then \@tempcnta := 15 fi
if \@tempcnta > 1 then \@tempcnta := \@tempcnta - 1 fi
\@tempcnta := \@tempcnta + CHAR
\@circlefnt
\char \@tempcnta
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

\if@ovt If producing the Top Bottom Left or Right of an oval.  
\if@ovb 427 \newif\if@ovt  
\if@ovl 428 \newif\if@ovb  
\if@ovr 429 \newif\if@ovl  
430 \newif\if@ovr

*(End of definition for \if@ovt and others.)*

```

\@ovxx
\@ovyy 431 \newdimen\@ovxx
\@ovdx 432 \newdimen\@ovyy
\@ovdy 433 \newdimen\@ovdx
\@ovro 434 \newdimen\@ovdy
\@ovri 435 \newdimen\@ovro
436 \newdimen\@ovri

```

*(End of definition for \@ovxx and others.)*

\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
437 \def\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
438 \@tempcnta\@tempdima
439 \@tempdima 4\p@\divide\@tempcnta\@tempdima
440 \ifnum \@tempcnta >10\relax
441 \@picture@warn
442 \@tempcnta 10\relax
443 \fi
444 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
```

Warn if requirements for oval or circle can't be met.

```

445 \else \@picture@warn \fi
446 \multiply\@tempcnta 4\relax
447 \setbox\@tempboxa \hbox{\@circlefnt
448 \char\@tempcnta}\@tempdima \wd\@tempboxa}
```

*(End of definition for \@getcirc.)*

\@picture@warn Generic warning for lines, vectors (used in \@sline) and oval or circle (used in \@getcirc) are not available at right size.

```

449 \def\@picture@warn{\@latex@warning{%
450 \string\oval, \string\circle, or \string\line\space
451 size unavailable}}
```

(End of definition for \@picture@warn.)

```
\@put
452 \def\@put#1#2#3{\raise #2\hb@xt@z@{\hspace{#1#3\hss}}}
(End of definition for \@put.)

\oval
453 \def\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2)[]}}
(End of definition for \oval.)

454 </2ekernel>
455 <latexrelease>\IncludeInRelease{2016/03/31}%
456 <latexrelease> {\@ovhlinetrue} %
457 <latexrelease> {Avoid almost zero length leaders} %
458 <*2ekernel | latexrelease>
```

\if@ovvline Tests whether horizontal or vertical lines are needed.

```
\if@ovhline
459 \newif\if@ovvline \ovvlinetrue
460 \newif\if@ovhline \ovhlinetrue
461 </2ekernel | latexrelease>
462 <latexrelease>\EndIncludeInRelease
463 <latexrelease>\IncludeInRelease{0000/00/00}%
464 <latexrelease> {\@ovhlinetrue} %
465 <latexrelease> {Avoid almost zero length leaders} %
466 <latexrelease>\let\if@ovvline\@undefined
467 <latexrelease>\let\if@ovhline\@undefined
468 <latexrelease>\EndIncludeInRelease
469 <*2ekernel>
```

(End of definition for \if@ovvline and \if@ovhline.)

```
\@oval
470 </2ekernel>
471 <*2ekernel | latexrelease>
472 <latexrelease>\IncludeInRelease{2020/10/01}%
473 <latexrelease> {\@oval}{default units} %
474 \def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
475 \ovttrue \ovbtrue \ovltrue \ovrtrue

476 \ovvlinetfalse \ovhlinetfalse
477 \otfor\reserved@a :=#3\do{ %
478 \csname @ov\reserved@a false\endcsname} %
479 \defaultunitsset\ovxx{#1}\unitlength
480 \defaultunitsset\ovyy{#2}\unitlength

481 \tempdimb \ifdim \ovyy >\ovxx \ovxx \ovvlinetrue
482 \else \ovyy \ifdim \ovyy =\ovxx \else \ovhlinetrue \fi \fi
483 \advance \tempdimb -2\p@
484 \getcirc \tempdimb
485 \ovro \ht\tempboxa \ovri \dp\tempboxa
486 \ovdx\ovxx \advance\ovdx -\tempdima \divide\ovdx \tw@
487 \ovdy\ovyy \advance\ovdy -\tempdima \divide\ovdy \tw@
```

```

488 \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
489 \ifdim \@ovdy >\z@ \@ovvlinetrue \fi
490 \@circlefnt \setbox\@tempboxa
491 \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
492 \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
493 \if@ovt \@ovhorz \kern -\@ovxx \fi
494 \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
495 \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
496 \c@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
497 \endgroup
498 (//ekernel | latexrelease)

499 \end{IncludeInRelease}
500 \end{IncludeInRelease}[2016/03/31]%
501 \oval{[\@oval]{default units}}%
502 \def\@oval(#1,#2)[#3]{\begin{group}\boxmaxdepth \maxdimen
503 \ovtrue \ovbtrue \ovltrue \ovrtrue
504 \ovlinefalse \ovhlinefalse
505 \tfor\reserved@a :=#3\do{%
506 \csname @ov\reserved@a false\endcsname}%
507 \ovxx #1\unitlength
508 \ovyy #2\unitlength
509 \tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
510 \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue
511 \fi\fi
512 \advance \@tempdimb -2\p@
513 \getcirc \@tempdimb
514 \ovro \ht\@tempboxa \ovri \dp\@tempboxa
515 \ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
516 \ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
517 \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
518 \ifdim \@ovdy >\z@ \@ovvlinetrue \fi
519 \@circlefnt \setbox\@tempboxa
520 \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
521 \if@ovl
522 \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
523 \fi
524 \if@ovt \@ovhorz \kern -\@ovxx \fi
525 \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
526 \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
527 \c@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
528 \endgroup
529 \end{IncludeInRelease}

530 \end{IncludeInRelease}[0000/00/00]%
531 \oval{[\@oval]{default units}}%
532 \def\@oval(#1,#2)[#3]{\begin{group}\boxmaxdepth \maxdimen
533 \ovtrue \ovbtrue \ovltrue \ovrtrue
534 \tfor\reserved@a :=#3\do
535 \csname @ov\reserved@a false\endcsname}%
536 \ovxx #1\unitlength
537 \ovyy #2\unitlength
538 \tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
539 \advance \@tempdimb -2\p@
540 \getcirc \@tempdimb

```

```

541 <|latexrelease> \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
542 <|latexrelease> \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
543 <|latexrelease> \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
544 <|latexrelease> \@circlefnt \setbox\@tempboxa
545 <|latexrelease> \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
546 <|latexrelease> \if@ovl
547 <|latexrelease> \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
548 <|latexrelease> \fi
549 <|latexrelease> \if@ovt \@ovhorz \kern -\@ovxx \fi
550 <|latexrelease> \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
551 <|latexrelease> \advance\@ovdy\@ovro \ht\@tempboxa\z@\dp\@tempboxa\z@
552 <|latexrelease> \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
553 <|latexrelease> \endgroup
554 <|latexrelease>\EndIncludeInRelease
555 <|2ekernel>

```

(End of definition for \oval.)

### \@ovvert

```

556 <|2ekernel>
557 <|latexrelease>\IncludeInRelease{2016/03/31}%
558 <|latexrelease> {\@ovvert}{Avoid almost zero length leaders}%
559 <|2ekernel | latexrelease>
560 \def\@ovvert#1#2{\vbox to\@ovyy{%
561 \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
562 \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
563 \else \kern \@ovri \kern \@ovdy \fi
564 \if@ovvline \leaders\vrule \@width \@wholewidth \fi
565 \vfil \nointerlineskip
566 \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
567 \hbox{\char \@tempcntb}%
568 \else \kern \@ovdy \kern \@ovro \fi}
569 <|2ekernel | latexrelease>
570 <|latexrelease>\EndIncludeInRelease
571 <|latexrelease>\IncludeInRelease{0000/00/00}%
572 <|latexrelease> {\@ovvert}{Avoid almost zero length leaders}%
573 <|latexrelease>\def\@ovvert#1#2{\vbox to\@ovyy{%
574 \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
575 \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
576 \else \kern \@ovri \kern \@ovdy \fi
577 \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
578 \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
579 \hbox{\char \@tempcntb}%
580 \else \kern \@ovdy \kern \@ovro \fi}
581 <|latexrelease>\EndIncludeInRelease
582 <|2ekernel>

```

(End of definition for \ovvert.)

### \@ovhorz

```

583 <|2ekernel>
584 <|latexrelease>\IncludeInRelease{2016/03/31}%
585 <|latexrelease> {\@ovhorz}{Avoid almost zero length leaders}%

```

```

586 {*2ekernel | latexrelease)
587 \def\@ovhorz{\hb@xt@0\@ovxx{\kern \c@ovro
588 \if@ovr \else \kern \c@ovdx \fi
589 \if@ovhline \leaders \hrule \c@height \c@wholewidth \fi
590 \hfil
591 \if@ovl \else \kern \c@ovdx \fi
592 \kern \c@ovri}}
593 {/2ekernel | latexrelease)
594 \end{IncludeInRelease}
595 \IncludeInRelease{0000/00/00}%
596 \end{latexrelease} {\c@ovhorz}{Avoid almost zero length leaders}%
597 \def\@ovhorz{\hb@xt@0\@ovxx{\kern \c@ovro
598 \if@ovr \else \kern \c@ovdx \fi
599 \leaders \hrule \c@height \c@wholewidth \hfil
600 \if@ovl \else \kern \c@ovdx \fi
601 \kern \c@ovri}}
602 \end{IncludeInRelease}
603 {*2ekernel}

```

*(End of definition for \@ovhorz.)*

```
\circle
604 \def\circle{\c@inmatherr\circle\@ifstar\@dot\@circle}
```

*(End of definition for \circle.)*

```
\@circle
605 {/2ekernel}
606 {*2ekernel | latexrelease)
607 \end{latexrelease} \IncludeInRelease{2020/10/01}%
608 \end{latexrelease} {\c@circle}{default units}%
609 \def\@circle#1{%
610 \begingroup \boxmaxdepth \maxdimen
611 \c@defaultunitsset\c@tempdimb{#1}\unitlength
612 \ifdim \c@tempdimb >15.5\p@ \getcirc\c@tempdimb
613 \c@ovro\ht\c@tempboxa
614 \setbox\c@tempboxa\hbox{\c@circlefont
615 \advance\c@tempcnta\tw@ \char \c@tempcnta
616 \advance\c@tempcnta\m@ne \char \c@tempcnta \kern -2\c@tempdima
617 \advance\c@tempcnta\tw@
618 \raise \c@tempdima \hbox{\char\c@tempcnta}\raise \c@tempdima
619 \box\c@tempboxa\ht\c@tempboxa\z@\dp\c@tempboxa\z@
620 \c@put{-\c@ovro}{-\c@ovro}{\box\c@tempboxa}%
621 \else \c@circ\c@tempdimb{96}\fi\endgroup
622 }{/2ekernel | latexrelease)
623 \end{IncludeInRelease}
624 \IncludeInRelease{0000/00/00}%
625 \end{latexrelease} {\c@circle}{default units}%
626 \def\@circle#1{%
627 \begingroup \boxmaxdepth \maxdimen \c@tempdimb #1\unitlength
628 \ifdim \c@tempdimb >15.5\p@ \getcirc\c@tempdimb
629 \c@ovro\ht\c@tempboxa
```

```

630 <|latexrelease> \setbox\@tempboxa\hbox{\@circlefnt
631 <|latexrelease> \advance\@tempcnta\tw@ \char \@tempcnta
632 <|latexrelease> \advance\@tempcnta\m@ne \char \@tempcnta
633 <|latexrelease> \kern -2\@tempdima
634 <|latexrelease> \advance\@tempcnta\tw@
635 <|latexrelease> \raise \@tempdima \hbox{\char\@tempcnta}%
636 <|latexrelease> \raise \@tempdima
637 <|latexrelease> \box\@tempboxa\ht\@tempboxa\z@ \dp\@tempboxa\z@
638 <|latexrelease> \put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
639 <|latexrelease> \else \circ\@tempdimb{96}\fi\endgroup
640 <|latexrelease>\EndIncludeInRelease
641 <|2ekernel>

```

(End of definition for `\@circle`.)

`\@dot` Internal form of `\circle*`.

```

642 <|2ekernel>
643 <|2ekernel | latexrelease>
644 <|latexrelease>\IncludeInRelease{2020/10/01}%
645 <|latexrelease> \{@dot\}{default units}%
646 \def\@dot#1{%
647 \@defaultunitsset\@tempdimb{#1}\unitlength
648 \circ\@tempdimb{112}%
649 <|2ekernel | latexrelease>
650 <|latexrelease>\EndIncludeInRelease
651 <|latexrelease>\IncludeInRelease{0000/00/00}%
652 <|latexrelease> \{@dot\}{default units}%
653 <|latexrelease>\def\@dot#1{\@tempdimb #1\unitlength \circ\@tempdimb{112}%
654 <|latexrelease>\EndIncludeInRelease
655 <|2ekernel>

```

(End of definition for `\@dot`.)

`\@circ`

```

656 \def\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
657 \@tempcnta\@tempdima \@tempdima \p@
658 \divide\@tempcnta\@tempdima
659 \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
660 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
661 \advance\@tempcnta #2\relax
662 \circ\@circlefnt \char\@tempcnta}

```

(End of definition for `\@circ`.)

`\@xarg` Counters used for manipulating the ‘slope’ arguments.

```

663 \newcount\@xarg
664 \newcount\@yarg
665 \newcount\@yyarg

```

(End of definition for `\@xarg`, `\@yarg`, and `\@yyarg`.)

`\@multicnt` Counter used in `\multiput`, and also `\multicolumn`.

```

666 \newcount\@multicnt

```

(End of definition for `\@multicnt`.)

```

\@xdim Length registers.
\@ydim 667 \newdimen\@xdim
668 \newdimen\@ydim

(End of definition for \@xdim and \@ydim.)

\@linechar Box for holding a line segment character, for sloping lines.
669 \newbox\@linechar

(End of definition for \@linechar.)

\@linelen Length of the line currently being built.
670 \newdimen\@linelen

(End of definition for \@linelen.)

\@clnwd Height and width of current line segment.
\@clnht 671 \newdimen\@clnwd
672 \newdimen\@clnht

(End of definition for \@clnwd and \@clnht.)

\@dashdim \dashbox internal registers.
\@dashbox 673 \newdimen\@dashdim
\@dashcnt 674 \newbox\@dashbox
675 \newcount\@dashcnt

(End of definition for \@dashdim, \@dashbox, and \@dashcnt.)
Initialization: “\thinlines”
676 \let\@linefnt\tenln
677 \let\@circlefnt\tencirc
678 \wholewidth\fontdimen8\tenln
679 \halfwidth .5\wholewidth

```

## 1.1 Curves

The new `\qbezier` command, based on the old `\bezier` defined in `bezier.sty`.  
*Historical L<sup>A</sup>T<sub>E</sub>X 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 \@xima := |BX - AX|
 \@xb := |CX - BX|
 \@xa := Max(\@xa, \@xb)
 \@ya := |BY - AY|
 \@yb := |CY - BY|
 \@ya := Max(\@ya, \@yb)
 @sc := Max(\@xa, \@ya)
 %% The coefficient .5 below is the degree of overlap of
 %% successive points, where 1 is no overlap and 0 is

```

```

%% complete overlap. A coefficient of C multiplies
%% the number of points plotted by 1/C.
%%
\@xa := .5 * \halfwidth
@sc := @sc / \halfwidth
@sc := Max(@sc, qbeziermax)
ELSE @sc := N
@scp := @sc+1
\@xb := 2 * (BX - AX) * \unitlength
\@xa := ((CX-AX)*\unitlength - \@xb)/@sc
\@yb := 2 * (BY - AY) * \unitlength
\@ya := ((CY-AY)*\unitlength - \@yb)/@sc
\@pictdot := square rule of width \wholewidth
\count@ := 0
WHILE \count@ < @scp
 DO \@xdim := ((\count@*\@xa + @xb) / @sc) * \count@
 \@ydim := ((\count@*\@ya + @yb) / @sc) * \count@
 plot pt with relative coords (\@xdim,\@ydim)
 \count@ := \count@+1
OD

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

\qbeziermax The maximum number of points to plot.

680 \def\qbeziermax{500}

*(End of definition for \qbeziermax.)*

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 (.

681 \newcommand\qbezier[2][0]{\bezier{#1}{#2}}

*(End of definition for \qbezier.)*

\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 (.

682 \def\bezier#1#2(#3)#4({\@bezier#1)(#3)()}

```

\@bezier 683 </2ekernel>
684 <*2ekernel | latexrelease>
685 <latexrelease>\IncludeInRelease{2020/10/01}%
686 <latexrelease> {\@bezier}{default units}%
687 \def\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
688 \ifnum #1=\z@%
689 \@defaultunitsset{\ovxx{#4}\unitlength
690 \@defaultunitsset{\advance\ovxx{-#2}\unitlength
691 \ifdim \ovxx<\z@ \ovxx -\ovxx \fi
692 \@defaultunitsset{\ovdx{#6}\unitlength
693 \@defaultunitsset{\advance\ovdx{-#4}\unitlength
694 \ifdim \ovdx<\z@ \ovdx -\ovdx \fi
695 \ifdim \ovxx<\ovdx \ovxx \ovdx \fi
696 \@defaultunitsset{\ovyy{#5}\unitlength
697 \@defaultunitsset{\advance\ovyy{-#3}\unitlength
698 \ifdim \ovyy<\z@ \ovyy -\ovyy \fi
699 \@defaultunitsset{\ovdy{#7}\unitlength
700 \@defaultunitsset{\advance\ovdy{-#5}\unitlength
701 \ifdim \ovdy<\z@ \ovdy -\ovdy \fi
702 \ifdim \ovyy<\ovdy \ovyy \ovdy \fi
703 \@multicnt
704 \ifdim \ovxx>\ovyy \ovxx \else \ovyy \fi
705 \ovxx .5\halfwidth \divide\@multicnt\ovxx
706 \ifnum \qbeziermax<\@multicnt
707 \@multicnt\qbeziermax\relax
708 \fi
709 \else \@multicnt#1\relax \fi
710 \tempcpta\@multicnt \advance\tempcpta\one
711 \@defaultunitsset{\ovdx{#4}\unitlength
712 \@defaultunitsset{\advance\ovdx{-#2}\unitlength
713 \multiply\ovdx\tw@
714 \@defaultunitsset{\ovxx{#6}\unitlength
715 \@defaultunitsset{\advance\ovxx{-#2}\unitlength
716 \advance\ovxx -\ovdx \divide\ovxx\@multicnt
717 \@defaultunitsset{\ovdy{#5}\unitlength
718 \@defaultunitsset{\advance\ovdy{-#3}\unitlength
719 \multiply\ovdy\tw@
720 \@defaultunitsset{\ovyy{#7}\unitlength
721 \@defaultunitsset{\advance\ovyy{-#3}\unitlength
722 \advance\ovyy -\ovdy \divide\ovyy\@multicnt
723 \setbox\tempboxa\hbox{%
724 \hspace{-\halfwidth}
725 \vrule \height\halfwidth
726 \depth \halfwidth
727 \width \wholewidth}%
728 \put(#2,#3){%
729 \count@\z@
730 \whilenum{\count@<\tempcpta}\do
731 {\xdim\count@\ovxx
732 \advance\xdim\ovdx
733 \divide\xdim\@multicnt
734 \multiply\xdim\count@

```

```

735 \@ydim\count@\@ovyy
736 \advance\@ydim\@ovdy
737 \divide\@ydim\@multicnt
738 \multiply\@ydim\count@
739 \raise \@ydim
740 \hb@xt@\z@{\kern\@xdim
741 \unhcopy\@tempboxa\hss}%
742 \advance\count@\@ne}}}
743 </2ekernel | latexrelease>
744 <| latexrelease>\EndIncludeInRelease
745 <| latexrelease>\IncludeInRelease{0000/00/00}%
746 <| latexrelease> {\@bezier}{default units}%
747 <| latexrelease>\def\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
748 <| latexrelease> \ifnum #1=\z@
749 <| latexrelease> \@ovxx #4\unitlength
750 <| latexrelease> \advance\@ovxx -#2\unitlength
751 <| latexrelease> \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
752 <| latexrelease> \@ovdx #6\unitlength
753 <| latexrelease> \advance\@ovdx -#4\unitlength
754 <| latexrelease> \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
755 <| latexrelease> \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
756 <| latexrelease> \@ovyy #5\unitlength
757 <| latexrelease> \advance\@ovyy -#3\unitlength
758 <| latexrelease> \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
759 <| latexrelease> \@ovdy #7\unitlength
760 <| latexrelease> \advance\@ovdy -#5\unitlength
761 <| latexrelease> \ifdim \@ovdy<\z@ \@ovdy -\@ovdy \fi
762 <| latexrelease> \ifdim \@ovyy<\@ovdy \@ovyy \@ovdy \fi
763 <| latexrelease> \@multicnt
764 <| latexrelease> \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
765 <| latexrelease> \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
766 <| latexrelease> \ifnum
767 <| latexrelease> \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax
768 <| latexrelease> \fi
769 <| latexrelease> \else \@multicnt#1\relax \fi
770 <| latexrelease> \@tempcnta\@multicnt \advance\@tempcnta\@ne
771 <| latexrelease> \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
772 <| latexrelease> \multiply\@ovdx \tw@
773 <| latexrelease> \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
774 <| latexrelease> \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
775 <| latexrelease> \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
776 <| latexrelease> \multiply\@ovdy \tw@
777 <| latexrelease> \@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
778 <| latexrelease> \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt
779 <| latexrelease> \setbox\@tempboxa\hbox{%
780 <| latexrelease> \hskip -\@halfwidth
781 <| latexrelease> \vrule \height\@halfwidth
782 <| latexrelease> \depth \@halfwidth
783 <| latexrelease> \width \@wholewidth}%
784 <| latexrelease> \put(#2,#3){%
785 <| latexrelease> \count@\z@
786 <| latexrelease> \whilenum{\count@<\@tempcnta}\do
787 <| latexrelease> {\@xdim\count@\@ovxx
788 <| latexrelease> \advance\@xdim\@ovdx

```

```

789 \divide\@xdim\@multicnt
790 \multiply\@xdim\count@
791 \ydim\count@\@ovyy
792 \advance\@ydim\@ovdy
793 \divide\@ydim\@multicnt
794 \multiply\@ydim\count@
795 \raise\@ydim
796 \hb@xt@{z}{\kern\@xdim}
797 \unhcopy\tempboxa\hss}%
798 \advance\count@\@ne}}}
799 \EndIncludeInRelease
800 {*2ekernel}

```

(End of definition for `\bezier` and `\obezier`.)

As the commands above all use “picture” interface we couldn’t define them with `\DeclareRobustCommand` so we do that now.

```

801 {/2ekernel}
802 {*2ekernel | latexrelease}
803 \IncludeInRelease{2019/10/01}%
804 \bezier{}{Make commands robust}%
805 \MakeRobust\bezier
806 \MakeRobust\circle
807 \MakeRobust\dashbox
808 \MakeRobust\line
809 \MakeRobust\linethickness
810 \MakeRobust\multiput
811 \MakeRobust\oval
812 \MakeRobust\put
813 \MakeRobust\qbezier
814 \MakeRobust\shortstack
815 \MakeRobust\thinlines
816 \MakeRobust\vector
817 {/2ekernel | latexrelease}
818 \IncludeInRelease
819 \IncludeInRelease{0000/00/00}%
820 \bezier{}{Make commands robust}%
821 \EndIncludeInRelease
822 \kernel@make@fragile\bezier
823 \kernel@make@fragile\circle
824 \kernel@make@fragile\dashbox
825 \kernel@make@fragile\line
826 \kernel@make@fragile\linethickness
827 \kernel@make@fragile\multiput
828 \kernel@make@fragile\oval
829 \kernel@make@fragile\put
830 \kernel@make@fragile\qbezier
831 \kernel@make@fragile\shortstack
832 \kernel@make@fragile\thinlines
833 \kernel@make@fragile\vector
834 \EndIncludeInRelease
835 {*2ekernel}
836 {/2ekernel}

```

# File 43

## 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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

#### DOCUMENT STYLE PARAMETERS

`\@thmcnter{COUNTER}` : A command such that  
`\edef\theCOUNTER{\@thmcnter{COUNTER}}`  
defines `\theCOUNTER` to produce a number for a theorem environment.  
The default is:

BEGIN `\noexpand\arabic{COUNTER}` END

`\@thmcntersep` : 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, `\@thmcntersep` should be `\def`'`. 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
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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X book indicates that \newtheorem may be used anywhere in the document...

```

1 {*2ekernel}
2 \def\newtheorem#1{%
3 \@ifnextchar[{\@othm{#1}}{\@nthm{#1}}}
```

(End of definition for \newtheorem.)

\@nthm

```

4 \def\@nthm#1#2{%
5 \@ifnextchar[{\@xnthm{#1}{#2}}{\@ynthm{#1}{#2}}}
```

(End of definition for \@nthm.)

\@xnthm 92/09/18 RmS: Changed \@addtoreset to \@newctr to produce error message if counter #3 does not exist (to be consistent with behaviour of \newcounter)

```

6 \def\@xnthm#1#2[#3]{%
7 \expandafter\@ifdefinable\csname #1\endcsname
8 {\@definecounter{#1}\@newctr{#1}[#3]\%
9 \expandafter\xdef\csname the#1\endcsname{%
10 \expandafter\noexpand\csname the#3\endcsname \@thmcOUNTERsep
11 \@thmcOUNTER{#1}}\%
12 \global\@namedef{#1}{\@thm{#1}{#2}}\%
13 \global\@namedef{end#1}{\@endtheorem}}}
```

(End of definition for \@xnthm.)

\@ynthm

```

14 \def\@ynthm#1#2{%
15 \expandafter\@ifdefinable\csname #1\endcsname
16 {\@definecounter{#1}\%
17 \expandafter\xdef\csname the#1\endcsname{\@thmcOUNTER{#1}}\%
18 \global\@namedef{#1}{\@thm{#1}{#2}}\%
19 \global\@namedef{end#1}{\@endtheorem}}}
```

(End of definition for \@ynthm.)

\@othm

```

20 \def\@othm#1[#2]{#3}{%
21 \@ifundefined{c@#2}{\@nocounterr{#2}}\%
22 {\expandafter\@ifdefinable\csname #1\endcsname
23 {\global\@namedef{the#1}{\@nameuse{the#2}}\%
24 \global\@namedef{#1}{\@thm{#2}{#3}}\%
25 \global\@namedef{end#1}{\@endtheorem}}}}
```

(End of definition for \@othm.)

\@thm

```

26 {/2ekernel}
27 {*2ekernel | latexrelease}
28 (latexrelease)\IncludeInRelease[2024/03/18]%
29 (latexrelease) {\@thm}{no link target}%
30 \def\@thm#1#2{%
```

```

31 \@kernel@refstepcounter{#1}%
32 \@ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
33 \langle latexrelease \rangle \EndIncludeInRelease
34 \langle latexrelease \rangle \IncludeInRelease{0000/00/00}%
35 \langle latexrelease \rangle \quad \{@thm}{no link target}%
36 \langle latexrelease \rangle \def\@thm#1#2{%
37 \langle latexrelease \rangle \quad \refstepcounter{#1}%
38 \langle latexrelease \rangle \@ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
39 \langle latexrelease \rangle \EndIncludeInRelease
40 \langle /2ekernel | latexrelease \rangle

```

(End of definition for `\@thm`.)

```

\@xthm
\@ythm
41 \langle *2ekernel \rangle
42 \def\@xthm#1#2{%
43 \begin{theorem}{#2}{\csname the#1\endcsname}\ignorespaces}
44 \def\@ythm#1#2[#3]{%
45 \opargbegintheorem{#2}{\csname the#1\endcsname}{#3}\ignorespaces}

```

(End of definition for `\@xthm` and `\@ythm`.)

Default values

```

\@thmcnter
\@thmcntersep
46 \def\@thmcnter#1{\noexpand\arabic{#1}}
47 \def\@thmcntersep{.}

```

(End of definition for `\@thmcnter` and `\@thmcntersep`.)

`\@begintheorem` Providing theorem defaults.

```

\@opargbegintheorem
\@endtheorem
48 \langle /2ekernel \rangle
49 \langle *2ekernel | latexrelease \rangle
50 \langle latexrelease \rangle \IncludeInRelease{2024/03/18}%
51 \langle latexrelease \rangle \quad \{@begintheorem}{add link targets}%
52 \def\@begintheorem#1#2{\trivlist
53 \item[\MakeLinkTarget{\currentcounter}\hspace{\labelsep}\bfseries #1\ #2]\itshape}
54 \def\@opargbegintheorem#1#2#3{\trivlist
55 \item[\MakeLinkTarget{\currentcounter}\hspace{\labelsep}\bfseries #1\ #2\ (#3)]\itshape}
56 \langle latexrelease \rangle \EndIncludeInRelease
57 \langle latexrelease \rangle \IncludeInRelease{0000/00/00}%
58 \langle latexrelease \rangle \quad \{@begintheorem}{add link targets}%
59 \langle latexrelease \rangle \def\@begintheorem#1#2{\trivlist
60 \item[\hspace{\labelsep}\bfseries #1\ #2]\itshape}
61 \langle latexrelease \rangle \def\@opargbegintheorem#1#2#3{\trivlist
62 \item[\hspace{\labelsep}\bfseries #1\ #2\ (#3)]\itshape}
63 \langle latexrelease \rangle \EndIncludeInRelease
64 \langle /2ekernel | latexrelease \rangle
65 \langle *2ekernel \rangle
66 \def\@endtheorem{\endtrivlist}
67 \langle /2ekernel \rangle

```

(End of definition for `\@begintheorem`, `\@opargbegintheorem`, and `\@endtheorem`.)

# File 44

## 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 L<sup>A</sup>T<sub>E</sub>X kernel.

The second part of the file defines the generic commands used for defining sectioning commands such as `\chapter`. Again the actual document level commands are defined in the class files, in terms of these commands.

```
1 {*2ekernel}
2 \message{title,}
```

#### 1.1 The Title

`\title` The user defines the title and author by the declarations `\title{<name>}`, `\author{<name>}`.  
`\author` Similarly the date is declared with `\date{<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 `\and` to be separated with the `\and` command.  
`\maketitle` And finally, the `\maketitle` command produces the actual title, using the information previously saved with the other commands.

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

`\title` `\title` for use in `\maketitle`. If not given `\maketitle` will produce an error message.  
7 `\DeclareRobustCommand\title[1]{\gdef\@title{\#1}}`

(End of definition for `\title`. This function is documented on page 751.)

`\author` `\author` for use in `\maketitle`. If not given `\maketitle` will produce a warning message.  
8 `\DeclareRobustCommand*\author[1]{\gdef\@author{\#1}}`

(End of definition for `\author`.)

`\date` `\date` for use in `\maketitle`. If not given `\maketitle` will produce `\today` as the default.

```
9 \DeclareRobustCommand*\date[1]{\gdef\@date{\#1}}
```

(End of definition for `\date`.)

`\thanks`  
10 `\DeclareRobustCommand\thanks[1]{\footnotemark}`  
11     `\protected\@xdef\@thanks{\@thanks`  
12         `\protect\footnotetext[\the\c@footnote]{\#1}}%`  
13 }

(End of definition for `\thanks`.)

```

\and
14 \DeclareRobustCommand{\and}{%
15 \end{tabular}%
16 \hskip 1em \oplus.17fil%
17 \begin{tabular}[t]{c}}% \end{tabular}

(End of definition for \and.)

18 </2ekernel | latexrelease>
19 <latexrelease>\EndIncludeInRelease
20 <latexrelease>\IncludeInRelease{0000/00/00}%
21 <latexrelease> {\title}{Make commands robust}%
22 <latexrelease>
23 <latexrelease>\kernel@make@fragile\title
24 <latexrelease>\kernel@make@fragile\author
25 <latexrelease>\kernel@make@fragile\date
26 <latexrelease>\kernel@make@fragile\thanks
27 <latexrelease>\kernel@make@fragile\and
28 <latexrelease>
29 <latexrelease>\EndIncludeInRelease
30 <*2ekernel>

\@title
31 \def\@title{\@latex@error{No \noexpand\title given}\@ehc}

(End of definition for \@title.)

\@author
32 \def\@author{\@latex@warning@no@line{No \noexpand\author given}{}}

(End of definition for \@author.)

\@date
33 \gdef\@date{\today}

(End of definition for \@date.)

\@thanks
34 \let\@thanks\empty

(End of definition for \@thanks.)

35 \message{sectioning,}

```

## 1.2 Sectioning

```

\@secpenalty
36 \newcount\@secpenalty
37 \@secpenalty = -300

(End of definition for \@secpenalty.)

\if@noskipsec Way back in 1991 (08/26) FMi & RmS set the \@noskipsec switch to true for the
\@noskipsectrue preamble and to false in \document. This was done to trap lists and related text in the
preamble but it does not catch everything.
38 \newif\if@noskipsec \@noskipsectrue

```

(End of definition for \if@noskipsec and \@noskipsectrue.)

\@startsection The \@startsection{\{name\}}{\{level\}}{\{indent\}}{\{beforeskip\}}{\{afterskip\}}{\{style\}}\*{\{altheading\}}{\{heading\}} command is the mother of all the user level sectioning commands. The part after the \*, including the \* is optional.

**name:** e.g., 'subsection'

**level:** a number, denoting depth of section – e.g., chapter = 0, section = 1, etc.

**indent:** Indentation of heading from left margin

**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 ...
 \par ...
 \endgroup}
```

Pseudocode for the \@startsection command *Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments* (not necessarily accurate any more):

```
\@startsection
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE} ==
BEGIN
 IF @noskipsec = T THEN \leavevmode FI
 % true if previous section had no body.

 \par
 @tempskipa := BEFORESKIP
 @afterindent := T
 IF @tempskipa < 0 THEN @tempskipa := -@tempskipa
 @afterindent := F
 FI
 IF @nobreak = true
 THEN \everypar == null
 ELSE \addpenalty{@secpenalty}
 \addvspace{@tempskipa}
 FI
 IF * next
```

```

 THEN \@ssect{INDENT}{BEForeskip}{AFTerskip}{Style}
 ELSE \@dblarg{\@sect
 {NAME}{LEVEL}{INDENT}
 {BEForeskip}{AFTerskip}{Style}}
 FI
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

39 \def\@startsection#1#2#3#4#5#6{%
40 \if@noskipsec \leavevmode \fi
41 \par
42 \tempskipa #4\relax
43 \if@afterindenttrue
44 \ifdim \tempskipa <\z@
45 \tempskipa -\tempskipa \if@afterindentfalse
46 \fi
47 \if@nobreak
48 \everypar{}%
49 \else
50 \addpenalty\secpenalty\addvspace\tempskipa
51 \fi
52 \ifstar
53 {\@ssect{#3}{#4}{#5}{#6}%
54 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}

```

*(End of definition for \@startsection.)*

\@sect Pseudocode for the \@sect command *Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```

\@sect{NAME}{LEVEL}
 {INDENT}{BEForeskip}{AFTerskip}
 {Style}[ARG1][ARG2]
 ==
BEGIN
 IF LEVEL > \c@secnumdepth
 THEN \svsec :=L null
 ELSE \refstepcounter{NAME}
 \svsec :=L BEGIN \secntformat{#1}\relax END
 FI
 IF AFTERSKIP > 0
 THEN \begingroup
 Style
 \hangfrom{\hskip INDENT\svsec}
 {\interlinepenalty 10000 ARG2\par}
 \endgroup
 \NAMEmark{ARG1}
 \addcontentsline{toc}{NAME}
 { IF LEVEL > \c@secnumdepth
 ELSE \protect\numberline{\theNAME} FI
 ARG1 }
 ELSE \svsechd == BEGIN Style
 \hskip INDENT\svsec

```

```

ARG2
\NAMEmark{ARG1}
\addcontentsline{toc}{NAME}
{ IF LEVEL > \c@secnumdepth
ELSE
 \protect\numberline{\theNAME}
FI
ARG1 }

END
FI
\@xsect{AFTERSKIP}
END
End of historical LATEX 2.09 comments.

55 \def\@sect#1#2#3#4#5#6[#7]#8{%
56 \ifnum #2>\c@secnumdepth
57 \let\@svsec\@empty
58 \else
59 \refstepcounter{#1}%

```

Since \secformat might end with an improper \hskip which is scanning forward for plus or minus we end the definition of \@svsec with \relax as a precaution.

```

60 \protected@edef\@svsec{\secformat{#1}\relax}%
61 \fi
62 \tempskipa #5\relax
63 \ifdim \tempskipa>\z@
64 \begingroup

```

This { used to be after the argument to \changefrom but was moved here to allow commands such as \MakeUppercase to be used at the end of #6.

```

65 #6{%
66 \changefrom{\hskip #3\relax\@svsec}%
67 \interlinepenalty \OM #8\@@par}%
68 \endgroup
69 \csname #1mark\endcsname{#7}%
70 \addcontentsline{toc}{#1}{%
71 \ifnum #2>\c@secnumdepth \else
72 \protect\numberline{\csname the#1\endcsname}%
73 \fi
74 #7}%
75 \else
\relax added 2 May 90

76 \def\@svsechd{%
77 #6{\hskip #3\relax
78 \@svsec #8}%
79 \csname #1mark\endcsname{#7}%
80 \addcontentsline{toc}{#1}{%
81 \ifnum #2>\c@secnumdepth \else
82 \protect\numberline{\csname the#1\endcsname}%
83 \fi
84 #7}%
85 \fi
86 \@xsect{#5}}

```

(End of definition for \@sect.)

\@xsect Pseudocode for the \@xsect command *Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
\@xsect{AFTERSKIP} ==
BEGIN
 IF AFTERSKIP > 0
 THEN \par \nobreak
 \vskip AFTERSKIP
 \afterheading
 ELSE @nobreak :=G F
 @noskipsec :=G T
 \everypar{ IF @noskipsec = T
 THEN @noskipsec :=G F
 \clubpenalty := 10000 % local
 \hskip -\parindent
 \begingroup
 \svsechd
 \endgroup
 \unskip
 \hskip -AFTERSKIP \relax
 %% relax added 14 Jan 91
 ELSE \clubpenalty := \clubpenalty % local
 \everypar := NULL
 FI
 }
 FI
END
```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```
87 \def\@xsect#1{%
88 \tempskipa #1\relax
89 \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.

```
90 \par \nobreak
91 \vskip \tempskipa
92 \afterheading
93 \else
94 \nobreakfalse
95 \global\noskipsectrue
96 \everypar{%
97 \ifnoskipsec
98 \global\noskipsecfalse
99 {\setbox\lastbox}\%
100 \clubpenalty\OM
101 \begingroup \svsechd \endgroup
102 \unskip
103 \tempskipa #1\relax
```

```

104 \hskip -\tempskipa
105 \else
106 \clubpenalty \clubpenalty
107 \everypar{}%
108 \fi}%
109 \fi
110 \ignorespaces}

```

(End of definition for \@xsect.)

\@seccntformat This command formats the section number including the space following it.

```

111 \def\@seccntformat#1{\csname the#1\endcsname\quad}

```

(End of definition for \@seccntformat.)

Pseudocode for the \@sect command *Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more)*:

```

\@sect{INDENT}{BEFRESKIP}{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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

Pseudocode for the \@afterheading command *Historical L<sup>A</sup>T<sub>E</sub>X 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 of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

\@ssect
112 \def\@ssect#1#2#3#4#5{%
113 \@tempskipa #3\relax
114 \ifdim \@tempskipa>\z@
115 \begingroup

```

This { used to be after the argument to `\@hangfrom` but was moved here to allow commands such as `\MakeUppercase` to be used at the end of #4.

```

116 #4{%
117 \@hangfrom{\hskip #1}%
118 \interlinepenalty \OM #5\@par}%
119 \endgroup
120 \else
121 \def\@svsechd{#4{\hskip #1\relax #5}}%
122 \fi
123 \@xsect{#3}

```

(End of definition for `\@ssect`.)

```

\if@afterindent
\@afterindenttrue
124 \newif\if@afterindent \@afterindenttrue

```

(End of definition for `\if@afterindent` and `\@afterindenttrue`.)

`\@afterheading` This hook is used in setting up custom-built headings in classes.dtx.

```

125 \def\@afterheading{%
126 \nobreaktrue
127 \everypar{%
128 \ifnobreak
129 \nobreakfalse
130 \clubpenalty \OM
131 \if@afterindent \else
132 {\setbox\z@\lastbox}%
133 \fi
134 \else
135 \clubpenalty \clubpenalty
136 \everypar{}%
137 \fi}}

```

(End of definition for `\@afterheading`.)

`\@hangfrom` `\@hangfrom{<text>}` : Puts `<text>` in a box, and makes a hanging indentation of the following material up to the first `\par`. Should be used in vertical mode.

```

138 \def\@hangfrom#1{\setbox\@tempboxa\hbox{#1}%
139 \hangindent \wd\@tempboxa\noindent\box\@tempboxa}

```

(End of definition for `\@hangfrom`.)

```

\c@secnumdepth
\c@tocdepth
140 \newcount\c@secnumdepth
141 \newcount\c@tocdepth

```

(End of definition for `\c@secnumdepth` and `\c@tocdepth`.)

```
\secdef \secdef{<unstarcmds>}{{<unstarcmds>}{<starcmds>}}
When defining a \chapter or \section command without using \startsection, you
can use \secdef as follows:
```

1. \def\chapter{ ... \secdef {\starcmd} {\unstarcmd} }
  2. \def{\starcmd}[#1]{#2}{...} % Command to define \chapter[...]{...}
  3. \def{\unstarcmd}{#1}{...} % Command to define \chapter\*{...}
- <sup>142</sup> \def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}

(End of definition for \secdef.)

### 1.2.1 Initializations

```
\sectionmark
\subsectionmark
\subsubsectionmark
\paragraphmark
\ subparagraphmark
143 \let\sectionmark\@gobble
144 \let\subsectionmark\@gobble
145 \let\subsubsectionmark\@gobble
146 \let\paragraphmark\@gobble
147 \let\ subparagraphmark\@gobble
```

(End of definition for \sectionmark and others.)

<sup>148</sup> \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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

\starttoc{EXT} ==

```

BEGIN
 \begingroup
 \makeatletter
 read file \jobname.EXT
 IF @filesw = true
 THEN open \jobname.EXT as file \tf@EXT
 FI
 @nobreak :=G FALSE %% added 24 May 89
 \endgroup
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

149 \def\@starttoc#1{%
150 \begingroup
151 \makeatletter
152 \cinput{\jobname.#1}%
153 \if@filesw
154 \expandafter\newwrite\csname tf@#1\endcsname
155 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
156 \fi
157 \nobreakfalse
158 \endgroup}

```

*(End of definition for \@starttoc.)*

\addcontentsline The \addcontentsline{\<table\>}{\<type\>}{\<entry\>} command allows the user to add his/her own entry to a table of contents, etc. The command adds the entry \contentsline{\<type\>}{\<entry\>}{\<page\>}{} to the .\<table\> file.

This macro is implemented as an application of \addtocontents. Note that \thepage is not expandable during \protected@write therefore one gets the page number at the time of the \shipout.

```

159 </2ekernel>
160 <2ekernel | latexrelease>
161 <latexrelease>\IncludeInRelease{2020/10/01}%
162 <latexrelease> {\addcontentsline}{fourth argument}%
163 \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.

```

164 \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}{}{}}%
165 \protected@file@percent}%
166 </2ekernel | latexrelease>
167 <latexrelease>\EndIncludeInRelease
168 <latexrelease>\IncludeInRelease{2018/12/01}%
169 <latexrelease> {\addcontentsline}{Mask line endings}%
170 <latexrelease> \def\addcontentsline#1#2#3{%
171 <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.

```

172 <latexrelease> \protected@file@percent}%
173 <latexrelease>\EndIncludeInRelease
174 <latexrelease>\IncludeInRelease{0000/00/00}%
175 <latexrelease> {\addcontentsline}{Mask line endings}%

```

```

176 \def\addcontentsline#1#2#3{%
177 \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}
178 \EndIncludeInRelease
179 {*2ekernel}

```

(End of definition for `\addcontentsline`.)

`\addtocontents` The `\addtocontents{<table>}{{<text>}}` command adds `<text>` to the `.<table>` file, with no page number.

```

180 \long\def\addtocontents#1#2{%
181 \protected@write\@auxout
182 {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
183 {\string\@writefile{#1}{#2}}}

```

(End of definition for `\addtocontents`.)

`\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, might be produced by:

```

\contentsline{subsection}
{\numberline{1.4.3}Gnats and Gnus}{22}{}

```

The `\protect` command causes command sequences to be written without expanding them.

```

184 {/2ekernel}
185 {*2ekernel | latexrelease}
186 \IncludeInRelease{2021/11/15}%
187 \contentsline{Four arguments}%

```

In the toc file `\contentsline` is followed by 4 arguments these days, but only the first 3 are used in the old interface. The fourth was by default empty and only used when `hyperref` was loaded. We now pick up all 4 arguments, save the last one away in `\@contentsline@destination` and then call the old interface. This is done to simplify the interface to `hyperref` and to prepare for future changes.

```

188 \def\contentsline#1#2#3#4{\gdef\@contentsline@destination{#4}%
189 \csname l@#1\endcsname{#2}{#3}}

```

Default definition.

```

190 \let\@contentsline@destination\empty
191 {/2ekernel | latexrelease}
192 \EndIncludeInRelease
193 \IncludeInRelease{0000/00/00}%
194 \contentsline{Four arguments}%
195 \let\@contentsline@destination\empty
196 \def\contentsline#1{\csname l@#1\endcsname}
197 \let\@contentsline@destination\undefined
198 \EndIncludeInRelease
199 {*2ekernel}

```

(End of definition for `\contentsline`.)

`\@dottedtocline{<level>}{{<indent>}}{<numwidth>} {{<title>}}{<page>}:` Macro to produce a table of contents line with the following parameters:

**level** If  $\langle level \rangle > \c@tocdepth$ , 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

```
200 {/2ekernel}
201 {*2ekernel | latexrelease}
202 {latexrelease}\IncludeInRelease{2018/12/01}%
203 {latexrelease} {\@dottedtocline}{Prevent protrusion}%
204 \def\@dottedtocline#1#2#3#4#5{%
205 \ifnum #1>\c@tocdepth \else
206 \vskip \z@ \oplus .2\p@
207 {\leftskip #2\relax \rightskip \c@tocrmarg \parfillskip -\rightskip
208 \parindent #2\relax\@afterindenttrue
209 \interlinepenalty\@M
210 \leavevmode
211 \tempdima #3\relax
212 \advance\leftskip \tempdima \null\nobreak\hskip -\leftskip
213 {#4}\nobreak
214 \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.

```
215 \mkern \dotsep mu\hbox{.}\mkern \dotsep
216 mu$\}\hfill
217 \nobreak
218 \hbox{\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.

```
219 \kern-\p@\kern\p@}%
220 \par}%
221 \fi}
```

(End of definition for \dottedtocline.)

- \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.

222 \DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}

(End of definition for \noprotrusion.)

```
223 </2ekernel | latexrelease>
224 <latexrelease>\EndIncludeInRelease
225 <latexrelease>\IncludeInRelease{0000/00/00}%
226 <latexrelease> {\@dottedtocline}{Prevent protrusion}%
227 <latexrelease>\def\@dottedtocline#1#2#3#4#5{%
228 <latexrelease> \ifnum #1>\c@tocdepth \else
229 <latexrelease> \vskip \z@ \oplus .2\p@
230 <latexrelease> {\leftskip #2\relax \rightskip \z@ \parfillskip -\rightskip
231 <latexrelease> \parindent #2\relax \afterindenttrue
232 <latexrelease> \interlinepenalty\OM
233 <latexrelease> \leavevmode
234 <latexrelease> \tempdima #3\relax
235 <latexrelease> \advance\leftskip \tempdima \null\nobreak\hskip -\leftskip
236 <latexrelease> {#4}\nobreak
237 <latexrelease> \leaders\hbox{$\m@th
238 <latexrelease> \mkern \z@ \dotsep \mu\hbox{.}\mkern \z@ \dotsep
239 <latexrelease> \mu$}\hfill
240 <latexrelease> \nobreak
241 <latexrelease> \hb@xt@\pnumwidth{\hfil\normalfont \normalcolor #5}%
242 <latexrelease> \par}%
243 <latexrelease> \fi}
244 <latexrelease>
245 <latexrelease>\let\noprotrusion\undefined
246 <latexrelease>\EndIncludeInRelease
247 </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{\langle number\rangle}: For use in a \contentsline command. It puts \langle number\rangle flush-left in a box of width \tempdima (Before 25 Jan 88 change, it also added \tempdima to the hanging indentation.)

248 \def\numberline#1{\hb@xt@\tempdima{\hfil}}
249 </2ekernel>

(End of definition for \numberline.)

# File 45

## 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

```
1 <*2ekernel>
2 \message{floats,}
```

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

Where floats may appear on a page, and how many may appear there are specified by the following float placement parameters. The numbers are named like counters so the user can set them with the ordinary counter-setting commands.

```
\c@topnumber : Number of floats allowed at the top of a column.
\topfraction : Fraction of column that can be devoted to floats.
\c@dbltopnumber, \dbltopfraction
 : Same as above, but for double-column floats.
\c@bottomnumber, \bottomfraction
 : Same as above for bottom of page.
\c@totalnumber : Number of floats allowed in a single column,
 including in-text floats.
{textfraction} : Minimum fraction of column that must contain text.
{floatpagefraction}: Minimum fraction of page that must be taken
 up by float page.
{dblfloatpagefraction}
 : Same as above, for double-column floats.
```

The document style must define the following.

```
\fps@TYPE : The default placement specifier for floats of type
 TYPE.

\ftype@TYPE : The type number for floats of type TYPE.

\ext@TYPE : The file extension indicating the file on which the
 contents list for float type TYPE is stored.
 For example, \ext@figure = 'lof'.

\fnum@TYPE : A macro to generate the figure number for a caption.
 For example, \fnum@TYPE == Figure \thefigure.

\@makecaption{NUM}{TEXT} :
 A macro to make a caption, with NUM the value
 produced by \fnum@... and TEXT the text of the caption.
 It can assume it's in a \parbox of the appropriate width.

\@float{TYPE}[PLACEMENT] : This macro begins a float environment for a
 single-column float of type TYPE with PLACEMENT as the placement
 specifier. The default value of PLACEMENT is defined by
 \fps@TYPE. The environment is ended by \end@float.
 E.g., \figure == \@float{figure}, \endfigure == \end@float.

\@float{TYPE}[PLACEMENT] ==
BEGIN
 if hmode then \@bsphack
 \@floatpenalty := -10002
 else \@floatpenalty := -10003
 fi
 \@capttype ==L TYPE
 \@dblflset
 \@fps ==L PLACEMENT
 \@onellevel@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{@capter}
\@dblarg{@caption{@capter}}
END

```

In following definition, `\par` moved from after `\addcontentsline` to before `\addcontentsline` because the `\write` could cause an extra blank line to be added to the paragraph above the caption. (Change made 12 Jun 87)

```

\@caption{TYPE}[STEXT]{TEXT} ==
BEGIN
\par
\addcontentsline{\ext@TYPE}{TYPE}{\numberline{\theTYPE}{STEXT}}
\begingroup
\@parboxrestore
\@normalsize
\@makecaption{\fnum@TYPE}{TEXT}
\par
\endgroup
END

```

`\@dblfloat{TYPE}[PLACEMENT]` : Macro to begin a float environment for a double-column float of type TYPE with PLACEMENT as the placement specifier. The default value of PLACEMENT is 'tp'. The environment is ended by `\end@dblfloat`.  
E.g., `\figure*` == `\@dblfloat{figure}`,  
`\endfigure*` == `\end@dblfloat`.

`\@dblfloat{TYPE}[PLACEMENT] ==`  
Identical to `\@float{TYPE}[PLACEMENT]` except `\hsize` and `\linewidth` are set to `\textwidth`.  
*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

`\@floatpenalty`  
<sup>3</sup> `\newcount\@floatpenalty`

(*End of definition for \@floatpenalty.*)

`\caption` This is set to be an error message outside a float since no `capttype` is defined there; this may need to be changed by some classes.

```

4 \def\caption{%
5 \ifx\@capttype\undefined
6 \@latex@error{\noexpand\caption outside float}\@ehd
7 \expandafter\@gobble
8 \else
9 \refstepcounter\@capttype
10 \expandafter\@firstofone
11 \fi
12 {\@dblarg{\@caption\@capttype}}%
13 }

```

(End of definition for `\caption`.)

`\@caption`

```

14 \long\def\@caption#1[#2]#3{%
15 \par
16 \addcontentsline{\csname ext@\#1\endcsname}{\#1}%
17 {\protect\numberline{\csname the#\#1\endcsname}{\ignorespaces #2}}%
18 \begingroup

```

The paragraph setting parameters are normalised at this point, however `\@parboxrestore` resets `\everypar` which is not correct in this context so `\@setminipage` is called if needed.

The float mechanism, like `minipage`, sets the flag `@minipage` true before executing the user-supplied text. Many L<sup>A</sup>T<sub>E</sub>X constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates T<sub>E</sub>X's 'top of page' behaviour. The flag must be set false at the start of the first paragraph. This is achieved by a redefinition of `\everypar`, but the call to `\@parboxrestore` removes that redefinition, so it is re-inserted if needed. If the flag is already false then the `\caption` was not the first entry in the float, and so some other paragraph has already activated the special `\everypar`. In this case no further action is needed.

```

19 \@parboxrestore
20 \if@minipage
21 \@setminipage
22 \fi
23 \normalsize
24 \@makecaption{\csname fnum@\#1\endcsname}{\ignorespaces #3}\par
25 \endgroup

```

(End of definition for `\@caption`.)

`\@float`

```

\@dblflset 26 \def\@float#1{%
27 \@ifnextchar[%]
28 {\@xflocat{\#1}}%
29 {\@edef\reserved@a{\noexpand\@xfloat{\#1}[\csname fps@\#1\endcsname]}%
30 \reserved@a}

```

(End of definition for `\@float` and `\@dblflset`.)

`\@dblfloat`

```

31 \def\@dblfloat{%
32 \if@twocolumn\let\reserved@a\@dbfl\else\let\reserved@a\@float\fi
33 \reserved@a

```

(End of definition for \@dblfloat.)

\fps@dbl Note that all double floats have default fps ‘tp’.

(End of definition for \fps@dbl.)

\@setfps This sets the fps, dealing with error conditions by adding the default.

(End of definition for \@setfps.)

\@xfloat The first part of this sets the count register that stores all the information about the type and fps of the float.

We assume here that the default specifiers already contain no active characters.

It may be better to store the defaults as numbers, rather than symbol strings.

```
34 </2ekernel>
35 <latexrelease>\IncludeInRelease{2015/01/01}%
36 <latexrelease> {\@xfloat}{Check float options}%
37 <2ekernel | latexrelease>
38 \def\@xfloat #1[#2]{%
39 \nодокумент
40 \def \@capttype {#1}%
41 \def \@fps {#2}%
42 \onelevel@sanitize \@fps
43 \def \reserved@b {!}%
44 \ifx \reserved@b \@fps
45 \@fpsadddefault
46 \else
47 \ifx \@fps \empty
48 \@fpsadddefault
49 \fi
50 \fi
51 \ifhmode
52 \bsphack
53 \@floatpenalty -\Mii
54 \else
55 \@floatpenalty-\Miii
56 \fi
57 \ifinner
58 \parmoderr\@floatpenalty\z@
59 \else
60 \next\currbox\freelist
61 {%
62 \tempcnta \sixt@n
63 \expandafter \tfor \expandafter \reserved@a
64 \expandafter :\expandafter =\@fps
65 \do
```

Start of changes, use a nested if structure, ending in an error.

```
66 {%
67 \if \reserved@a h%
68 \ifodd \tempcnta
69 \else
70 \advance \tempcnta \one
71 \fi
```

```

72 \else\if \reserved@a t%
73 \@setfpsbit \tw@%
74 \else\if \reserved@a b%
75 \@setfpsbit 4%
76 \else\if \reserved@a p%
77 \@setfpsbit 8%
78 \else\if \reserved@a !%
79 \ifnum \tempcnta>15
80 \advance\tempcnta -\sixt@@n\relax
81 \fi
82 \else
83 \@latex@error{Unknown float option `\'\reserved@a'}%
84 {Option `\'\reserved@a' ignored and `p' used.}%
85 \@setfpsbit 8%
86 \fi\fi\fi\fi\fi
87 }%

```

End of changes

```

88 \tempcntb \csname ftype@\@capttype \endcsname
89 \multiply \tempcntb \xxxii
90 \advance \tempcnta \tempcntb
91 \global \count\currbox \tempcnta
92 }%
93 \fltofvf
94 \fi

```

The remainder sets up the box in which the float is typeset, and the typesetting environment to be used. It is essential to have the extra box to avoid the unwanted space that would otherwise often be put at the top of the float.

It ends with a hook; not sure how useful this is but it is needed at present to deal with double-column floats.

```

95 \global \setbox\currbox
96 \color@vbox
97 \normalcolor
98 \vbox \bgroup
99 \hsize\columnwidth
100 \parboxrestore
101 \floatboxreset
102 }%
103 </2ekernel | latexrelease>
104 <latexrelease>\EndIncludeInRelease
105 <latexrelease>\IncludeInRelease{0000/00/00}%
106 <latexrelease> {\@xfloat}{Check float options}%
107 <latexrelease>\def\@xfloat #1[#2]{%
108 <latexrelease> \noldocument
109 <latexrelease> \def \@capttype {#1}%
110 <latexrelease> \def \fps {#2}%
111 <latexrelease> \onelevel@sanitize \fps
112 <latexrelease> \def \reserved@b {!}%
113 <latexrelease> \ifx \reserved@b \fps
114 <latexrelease> \fpsadddefault
115 <latexrelease> \else
116 <latexrelease> \ifx \fps \empty
117 <latexrelease> \fpsadddefault

```

```

118 <|latexrelease> \fi
119 <|latexrelease> \fi
120 <|latexrelease> \ifhmode
121 <|latexrelease> \@bsphack
122 <|latexrelease> \@floatpenalty -\@Mii
123 <|latexrelease> \else
124 <|latexrelease> \@floatpenalty-\@Miii
125 <|latexrelease> \fi
126 <|latexrelease> \ifinner
127 <|latexrelease> \@parmoderr\@floatpenalty\z@\@parmoderr
128 <|latexrelease> \else
129 <|latexrelease> \@next\@currbox\@freelist
130 <|latexrelease> {%
131 <|latexrelease> \@tempcnta \sixt@@n
132 <|latexrelease> \expandafter \@tfor \expandafter \reserved@a
133 <|latexrelease> \expandafter :\expandafter =\@fps
134 <|latexrelease> \do
135 <|latexrelease> {%
136 <|latexrelease> \if \reserved@a h%
137 <|latexrelease> \ifodd \@tempcnta
138 <|latexrelease> \else
139 <|latexrelease> \advance \@tempcnta \one
140 <|latexrelease> \fi
141 <|latexrelease> \fi
142 <|latexrelease> \if \reserved@a t%
143 <|latexrelease> \@setfpsbit \tw@
144 <|latexrelease> \fi
145 <|latexrelease> \if \reserved@a b%
146 <|latexrelease> \@setfpsbit 4%
147 <|latexrelease> \fi
148 <|latexrelease> \if \reserved@a p%
149 <|latexrelease> \@setfpsbit 8%
150 <|latexrelease> \fi
151 <|latexrelease> \if \reserved@a !%
152 <|latexrelease> \ifnum \@tempcnta>15
153 <|latexrelease> \advance\@tempcnta -\sixt@@n\relax
154 <|latexrelease> \fi
155 <|latexrelease> \fi
156 }%
157 <|latexrelease> \@tempcntb \csname ftype@\@capttype \endcsname
158 <|latexrelease> \multiply \@tempcntb \xxxii
159 <|latexrelease> \advance \@tempcnta \@tempcntb
160 <|latexrelease> \global \count\@currbox \@tempcnta
161 }%
162 <|latexrelease> \@fltovf
163 <|latexrelease> \fi
164 <|latexrelease> \global \setbox\@currbox
165 <|latexrelease> \color@vbox
166 <|latexrelease> \normalcolor
167 <|latexrelease> \vbox \bgroup
168 <|latexrelease> \hsize\columnwidth
169 <|latexrelease> \parboxrestore
170 <|latexrelease> \floatboxreset
171 <|latexrelease>}%

```

```

172 ⟨latexrelease⟩\EndIncludeInRelease
173 ⟨*2ekernel⟩

(End of definition for \@xfloat.)
```

\@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 \setnobreak; otherwise this command will be redundant.

```

174 \def \@floatboxreset {%
175 \reset@font
176 \normalsize
177 \setminipage
178 }
```

(End of definition for \@floatboxreset.)

```
\@setnobreak
179 \def \@setnobreak{%
180 \if@nobreak
181 \let\outer@nobreak\@nobreaktrue
182 \else
183 \fi
184 }
```

(End of definition for \@setnobreak.)

```
\@setminipage
185 \def \@setminipage{%
186 \minipagetrue
187 \everypar{\minipagefalse\everypar{}}
188 }
```

(End of definition for \@setminipage.)

```
\end@float
189 \def\end@float{%
190 \endfloatbox
191 \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed \textheight, otherwise float will never get typeset (91/03/15 FMi).

```

192 \largefloatcheck
193 \cons\currlist\currbox
194 \ifnum\@floatpenalty <-\@Mii
195 \penalty -\@Miv
```

Saving and restoring \prevdepth added 26 May 87 to prevent extra vertical space when used in vertical mode.

```

196 \tempdima\prevdepth
197 \vbox{}%
198 \prevdepth\tempdima
```

```

199 \penalty\@floatpenalty
200
201 \else
202 \vadjust{\penalty -\@Miv \vbox{} \penalty\@floatpenalty}\@EspHack
203 \fi
204 \fi
205 }

```

(End of definition for \end@float.)

\end@dblfloat

```

205 </2ekernel>
206 <|latexrelease|\IncludeInRelease{2015/01/01}%
207 <|latexrelease| {\end@dblfloat}{float order in 2-column}%
208 <*2ekernel | latexrelease>
209 \def\end@dblfloat{%
210 \if@twocolumn
211 \endfloatbox
212 \ifnum\@floatpenalty <\z@
213 \largefloatcheck

```

Force the depth of two column float boxes.

```
214 \global\dp\@currbox1sp %
```

What follows is essentially \end@float without a starting \endfloatbox.

```

215 \cons\@currlist\@currbox
216 \ifnum\@floatpenalty <-\@Mi
217 \penalty -\@Miv
218 \tempdima\prevdepth
219 \vbox{}%
220 \prevdepth\tempdima
221 \penalty\@floatpenalty
222 \else
223 \vadjust{\penalty -\@Miv \vbox{} \penalty\@floatpenalty}\@EspHack
224 \fi
225
226 \fi
227 \else
228 \end@float
229 \fi
230 }%
231 </2ekernel | latexrelease>
232 <|latexrelease|\EndIncludeInRelease
233 <|latexrelease|\IncludeInRelease{0000/00/00}%
234 <|latexrelease|\def\end@dblfloat{%
235 <|latexrelease|\if@twocolumn
236 <|latexrelease| \endfloatbox
237 <|latexrelease| \ifnum\@floatpenalty <\z@

```

We make sure that we never exceed \textheight, otherwise float will never get typeset (91/03/15 FMI).

```

238 <|latexrelease| \largefloatcheck
239 <|latexrelease| \cons\@dbldeferlist\@currbox
240 <|latexrelease| \fi

```

RmS 92/03/18 changed \c@esphack to \c@Ephack.

```
241 〈\textrlease〉 \ifnum \c@floatpenalty =-\c@Mii \c@Ephack\fi
242 〈\textrlease〉\else
243 〈\textrlease〉 \end\float
244 〈\textrlease〉\fi
245 〈\textrlease〉} %
246 〈\textrlease〉\EndIncludeInRelease
247 〈*2ekernel〉
```

(End of definition for \enddblfloat.)

\c@endfloatbox This macro is not intended to be a hook; it is designed to help maintain the integrity of this code, which is used twice and, as can be seen, is subject to frequent changes.

```
248 \def \c@endfloatbox{ %
249 \par\vskip\z@skip %% \par\vskip\z@ added 15 Dec 87
250 \c@minipagefalse
251 \outer\nobreak
252 \egroup %% end of vbox
253 \color@endbox
254 }
```

(End of definition for \c@endfloatbox.)

\outer\nobreak

```
255 \let\outer\nobreak\empty
```

(End of definition for \outer\nobreak.)

\c@largefloatcheck This calculates by how much a float is oversize for the page and prints this in a warning message.

```
256 \def \c@largefloatcheck{ %
257 \ifdim \ht\c@currbox>\textheight
258 \c@tempdima -\textheight
259 \advance \c@tempdima \ht\c@currbox
260 \c@latex@warning {Float too large for page by \the\c@tempdima} %
261 \ht\c@currbox \textheight
262 \fi
263 }
```

(End of definition for \c@largefloatcheck.)

```
\c@dbf1t
\c@dblfloat
264 \def\c@dbf1t#1{\c@ifnextchar[{\c@dblfloat{#1}}{\c@dblfloat{#1}[tp]}}
265 \def\c@dblfloat#1[#2]{%
266 \c@xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}
```

(End of definition for \c@dbf1t and \c@dblfloat.)

Moved to ltoutput 93/12/16

```
267 \%newcount\c@topnumber
268 \%newcount\c@dbltopnumber
269 \%newcount\c@bottomnumber
270 \%newcount\c@totalnumber
```

|                     |                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \@floatplacement    | An analysis of \@floatplacement:<br>This should be called whenever \@colht has been set.                                                                                                                                                                                                                                                                                                        |
|                     | <pre> 271 \def\@floatplacement{\global\@topnum\c@topnumber 272   % Textpage bit, global: 273   \global\@toproom \topfraction\@colht 274   \global\@botnum \c@bottomnumber 275   \global\@botroom \bottomfraction\@colht 276   \global\@colnum \c@totalnumber 277   % Floatpage bit, local: 278   \@fpmin \floatpagefraction\@colht} 279 </pre> <p>(End of definition for \@floatplacement.)</p> |
| \@dblfloatplacement | This should be called only within a group. Now changed to provide extra checks in \@addtoblcol, needed when processing a BANG float.                                                                                                                                                                                                                                                            |
|                     | <pre> 280 &lt;texreleas&gt;\IncludeInRelease{2015/01/01}% 281 &lt;texreleas&gt;      {\@dblfloatplacement}{float order in 2-column}% 282 </pre>                                                                                                                                                                                                                                                 |
|                     | When making two column float area, look for floats with 1sp depth.                                                                                                                                                                                                                                                                                                                              |
|                     | <pre> 283 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber 284   \global\@dbltoproom \dbltopfraction\@colht 285   \@textmin \@colht 286   \advance \@textmin -\@dbltoproom 287   \@fpmin \dblfloatpagefraction\textheight 288   \@fptop \@dblftop 289   \@fpsep \@dblfpsep 290   \@fpbot \@dblfpbot 291 </pre>                                                                         |
|                     | \f@depth is used in \@testwrongwidth to look for either column or dbl-column floats. A value of 1sp signals the latter. Because of this setting here, \@dblfloatplacement needs to be called inside a group which is a questionable design.                                                                                                                                                     |
|                     | <pre> 292 </pre>                                                                                                                                                                                                                                                                                                                                                                                |
|                     | Textpage bit: global, but need not be.                                                                                                                                                                                                                                                                                                                                                          |
|                     | <pre> 297 &lt;texreleas&gt; \global\@dbltopnum\c@dbltopnumber 298 &lt;texreleas&gt; \global\@dbltoproom \dbltopfraction\@colht 299 </pre>                                                                                                                                                                                                                                                       |
|                     | This new bit uses \@textmin to locally store the amount of extra room in the column.                                                                                                                                                                                                                                                                                                            |
|                     | <pre> 300 &lt;texreleas&gt; \@textmin \@colht 301 &lt;texreleas&gt; \advance \@textmin -\@dbltoproom 302 </pre>                                                                                                                                                                                                                                                                                 |
|                     | Floatpage bit: must be local.                                                                                                                                                                                                                                                                                                                                                                   |
|                     | <pre> 303 &lt;texreleas&gt; \@fpmin \dblfloatpagefraction\textheight 304 &lt;texreleas&gt; \@fptop \@dblftop 305 &lt;texreleas&gt; \@fpsep \@dblfpsep 306 &lt;texreleas&gt; \@fpbot \@dblfpbot 307 </pre>                                                                                                                                                                                       |

(End of definition for \dblfloatplacement.)

Historical L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

\marginpar
308 \def\marginpar{%
309 \ifhmode
310 \bsphack
311 \floatpenalty -\Mii
312 \else
313 \floatpenalty-\Miii
314 \fi
315 \ifinner
316 \parmoderr
317 \floatpenalty\z@
318 \else
319 \next\@currbox\@freelist{}{%
320 \next\@marbox\@freelist{\global\count\@marbox\m@ne}%
321 {\floatpenalty\z@
322 \f@tovf\def\@currbox{\tempboxa}\def\@marbox{\tempboxa}}%
323 \fi
324 \ifnextchar [\@xmpar\@ympar}

```

*(End of definition for \marginpar.)*

```

\@xmpar
325 \long\def\@xmpar[#1]{%
326 \@savemarbox\@marbox{#1}%
327 \@savemarbox\@currbox{#2}%
328 \@xympar}

```

*(End of definition for \@xmpar.)*

```

\@ympar
329 \long\def\@ympar#1{%
330 \@savemarbox\@marbox{#1}%
331 \global\setbox\@currbox\copy\@marbox
332 \@xympar}

```

*(End of definition for \@ympar.)*

```

\@savemarbox
 333 </2ekernel>
 334 {*2ekernel | latexrelease}
 335 <latexrelease>\IncludeInRelease{2021/06/01}%
 336 <latexrelease> {\@savemarbox}{Explicit par for marginpar}%
 337 \long\def \@savemarbox #1#2{%
 338 \global\setbox #1%
 339 \color@vbox
 340 \vtop{%
 341 \hsize\marginparwidth
 342 \parboxrestore
 343 \marginparreset
 344 #2\par
 345 \minipagetrue
 346 \outer@nobreak
 347 }%
 348 \color@endbox
 349 }
 350 </2ekernel | latexrelease>
 351 <latexrelease>\EndIncludeInRelease
 352 <latexrelease>\IncludeInRelease{0000/00/00}%
 353 <latexrelease> {\@savemarbox}{Explicit par for marginpar}%
 354 <latexrelease>
 355 <latexrelease>\long\def \@savemarbox #1#2{%
 356 \global\setbox #1%
 357 \color@vbox
 358 \vtop{%
 359 \hsize\marginparwidth
 360 \parboxrestore
 361 \marginparreset
 362 #2%
 363 \minipagetrue
 364 \outer@nobreak
 365 }%
 366 \color@endbox
 367 <latexrelease>}
 368 <latexrelease>\EndIncludeInRelease
 369 {*2ekernel}

```

(End of definition for `\@savemarbox`.)

`\marginparreset` 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 `\setnobreak`; otherwise this command will be redundant.

```

370 \def \marginparreset {%
371 \reset@font
372 \normalsize
373 \% \let\if@nobreak\iffalse
374 \% \let\if@noskipsec\iffalse

```

```

375 %
376 \@setnobreak
377 \@setminipage
378 }

```

(End of definition for \marginparreset.)

### \@xympar

Setting the box here is done only because the code uses \end@float; it will be empty and gets discarded.

```

378 \def \@xympar{%
379 \ifnum\@floatpenalty <\z@\@cons\@currlist\@marbox\fi
380 \setbox\@tempboxa
381 \color@vbox
382 \vbox \bgroup
383 \end@float
384 \ignorespacesfalse
385 \esphack
386 }

```

(End of definition for \@xympar.)

### \reversemarginpar \normalmarginpar

```

387 \def\reversemarginpar{\global\@mparbottom\z@\@reversemargintrue}
388 \def\normalmarginpar{\global\@mparbottom\z@\@reversemarginfalse}

```

(End of definition for \reversemarginpar and \normalmarginpar.)

```
389 \message{footnotes,}
```

## 1.2 Footnotes

*Historical L<sup>A</sup>T<sub>E</sub>X 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 skip to a negative value 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  
 $\text{\textsuperscript}{\@thefnmark}$

**\@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  
 $\$^{\@thefnmark}\$ \text{ NOTE}$

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 \c@sf := \the\spacefactor FI
\@makefnmark % put number in main text
IF hmode THEN \spacefactor := \c@sf FI
END

\footnotetext ==
BEGIN begingroup \protect == \noexpand
\@thefnmark :=G eval (\thempfn)
endgroup
\@footnotetext
END

\footnotetext[NUM] ==
BEGIN begingroup counter \c@mpfn :=L NUM
\protect == \noexpand
\@thefnmark :=G eval (\thempfn)
endgroup
\@footnotetext
END

```

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

**\footins** L<sup>A</sup>T<sub>E</sub>X does use the same insert for footnotes as PLAIN.

390 \newinsert\footins  
L<sup>A</sup>T<sub>E</sub>X leaves these initializations for the \footins insert.

391 \skip\footins=\bigskipamount % space added when footnote is present  
392 \count\footins=1000 % footnote magnification factor (1 to 1)  
393 \dimen\footins=8in % maximum footnotes per page

*(End of definition for \footins.)*

**\footnoterule** L<sup>A</sup>T<sub>E</sub>X keeps PLAIN T<sub>E</sub>X's \footnoterule as the default.

394 \def\footnoterule{\kern-3\p@  
395 \hrule \width 2in \kern 2.6\p@} % the \hrule is .4pt high

*(End of definition for \footnoterule.)*

**\thefootnote**

396 \@definecounter{footnote}  
397 \def\thefootnote{\@arabic\c@footnote}

*(End of definition for \thefootnote.)*

**\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.

398 \Qdefinecounter{mpfootnote}  
399 \def\thempfootnote{\itshape\@alph\c@mpfootnote}}

*(End of definition for \thempfootnote.)*

\@makefnmark Default definition.

```

400 \%def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
401 \def\@makefnmark{\hbox{\normalfont\@thefnmark}}

```

(End of definition for \@makefnmark.)

\textsuperscript This command provides superscript characters in the current text font. It's implementation might change!!!

```

402 \DeclareRobustCommand*\textsuperscript[1]{%
403 \selectfont#1}

```

(End of definition for \textsuperscript.)

\@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.

```

404 (/2ekernel)
405 <*2ekernel | latexrelease>
406 <latexrelease>\IncludeInRelease{2020/10/01}%
407 <latexrelease> {\@textsuperscript}{superscript baseline}%
408 \def\@textsuperscript#1{%
409 {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\sf@size#1}}}}}
410 (/2ekernel | latexrelease)
411 <latexrelease>\EndIncludeInRelease
412 <latexrelease>\IncludeInRelease{0000/00/00}%
413 <latexrelease> {\@textsuperscript}{superscript baseline}%
414 <latexrelease>
415 <latexrelease>\def\@textsuperscript#1{%
416 <latexrelease> {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}}
417 <latexrelease>\EndIncludeInRelease
418 <*2ekernel>

```

(End of definition for \@textsuperscript.)

\textsubscript

```

419 (/2ekernel)
420 <latexrelease>\IncludeInRelease{2015/01/01}%
421 <latexrelease> {\textsubscript}{\textsubscript}%
422 <*2ekernel | latexrelease>
423 \DeclareRobustCommand*\textsubscript[1]{%
424 \textsubscript{\selectfont#1}}%
425 (/2ekernel | latexrelease)
426 <latexrelease>\EndIncludeInRelease
427 <latexrelease>\IncludeInRelease{0000/00/00}%
428 <latexrelease> {\textsubscript}{\textsubscript}%
429 <latexrelease>\let\textsubscript@\undefined
430 <latexrelease>\EndIncludeInRelease
431 <*2ekernel>

```

(End of definition for \textsubscript.)

```

\@textsubscript
 432 </2ekernel>
 433 {*2ekernel | latexrelease}
 434 <latexrelease>\IncludeInRelease{2020/10/01}%
 435 <latexrelease> {\@textsubscript}{subscript baseline}%
 436 \def\@textsubscript#1{%
 437 {\m@th\ensuremath{_f\mbox{\scriptsize\sffamily\sf@size\sf@size#1}}}}
 438 </2ekernel | latexrelease>
 439 <latexrelease>\EndIncludeInRelease
 440 <latexrelease>\IncludeInRelease{2015/01/01}%
 441 <latexrelease> {\@textsubscript}{subscript baseline}%
 442 <latexrelease>
 443 <latexrelease>\def\@textsubscript#1{%
 444 {\m@th\ensuremath{_f\mbox{\scriptsize\sffamily\sf@size\z@#1}}}}
 445 <latexrelease>\EndIncludeInRelease
 446 <latexrelease>\IncludeInRelease{0000/00/00}%
 447 <latexrelease> {\@textsubscript}{subscript baseline}%
 448 <latexrelease>\let\@textsubscript\undefined
 449 <latexrelease>\EndIncludeInRelease
 450 {*2ekernel}

(End of definition for \@textsubscript.)
```

\footnotesep

```

 451 \newdimen\footnotesep
```

(End of definition for \footnotesep.)

\footnote

```

 452 \def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
 453 \protected@xdef\@thefnmark{\thempfn}%
 454 \@footnotemark\@footnotetext}}
```

(End of definition for \footnote.)

\@xfootnote

```

 455 \def\@xfootnote[#1]{%
 456 \begingroup
 457 \csname c@\@mpfn\endcsname #1\relax
 458 \unrestored@protected@xdef\@thefnmark{\thempfn}%
 459 \endgroup
 460 \@footnotemark\@footnotetext}
```

(End of definition for \@xfootnote.)

\@footnotetext

```

 461 </2ekernel>
 462 {*2ekernel | latexrelease}
 463 <latexrelease>\IncludeInRelease{2021/11/15}%
 464 <latexrelease> {\@footnotetext}{footnotetext tagging}%
 465 \long\def\@footnotetext#1{\insert\footins{%
 466 \reset@font\footnotesize
 467 \interlinepenalty\interfootnotelinepenalty
 468 \splittopskip\footnotesep}}
```

```

469 \splitmaxdepth \dp\strutbox \floatingpenalty \OMM
470 \hsize\columnwidth \parboxrestore
471 \def\@currentcounter{footnote}%
472 \protected@edef\@currentlabel{%
473 \csname p@footnote\endcsname\thefnmark
474 }%
475 \color@begingroup
476 \makefntext{%
477 \rule{z@\footnotesep\ignorespaces#1\finalstrut\strutbox}%
478 \par
479 \color@endgroup}%
480 {/2ekernel | latexrelease}
481 \end{IncludeInRelease}

482 \begin{IncludeInRelease}[2021/06/01]
483 \begin{latexrelease}
484 {\@footnotetext}{footnotetext tagging}%
485 \long\def\@footnotetext#1{\insert\footins{%
486 \reset@font\footnotesize
487 \interlinepenalty\interfootnotelinepenalty
488 \splittopskip\footnotesep
489 \splitmaxdepth \dp\strutbox \floatingpenalty \OMM
490 \hsize\columnwidth \parboxrestore
491 \protected@edef\@currentlabel{%
492 \csname p@footnote\endcsname\thefnmark
493 }%
494 \color@begingroup
495 \makefntext{%
496 \rule{z@\footnotesep\ignorespaces#1\finalstrut\strutbox}%
497 \par
498 \color@endgroup}%
499 \end{IncludeInRelease}
500 \begin{IncludeInRelease}[0000/00/00]
501 \begin{latexrelease}
502 {\@footnotetext}{footnotetext tagging}%
503 \long\def\@footnotetext#1{\insert\footins{%
504 \reset@font\footnotesize
505 \interlinepenalty\interfootnotelinepenalty
506 \splittopskip\footnotesep
507 \splitmaxdepth \dp\strutbox \floatingpenalty \OMM
508 \hsize\columnwidth \parboxrestore
509 \protected@edef\@currentlabel{%
510 \csname p@footnote\endcsname\thefnmark
511 }%
512 \color@begingroup
513 \makefntext{%
514 \rule{z@\footnotesep\ignorespaces#1\finalstrut\strutbox}%
515 \color@endgroup}%
516 \end{IncludeInRelease}
517 {*2ekernel}

```

(End of definition for \footnotetext.)

\footnotemark

```

518 \def\footnotemark{%
519 \ifnextchar[\@xfootnotemark
520 {\stepcounter{footnote}%
521 \protected@xdef\@thefnmark{\thefootnote}%
522 \@footnotemark}%

```

(End of definition for `\footnotemark`.)

`\@xfootnotemark`

```

523 \def\@xfootnotemark[#1]{%
524 \begingroup
525 \c@footnote #1\relax
526 \unrestored@protected@xdef\@thefnmark{\thefootnote}%
527 \endgroup
528 \@footnotemark}%

```

(End of definition for `\@xfootnotemark`.)

`\@footnotemark`

```

529 \def\@footnotemark{%
530 \leavevmode
531 \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
532 \makefnmark
533 \ifhmode\spacefactor\@x@sf\fi
534 \relax}%

```

(End of definition for `\@footnotemark`.)

`\footnotetext`

```

535 \def\footnotetext{%
536 \ifnextchar [\@xfootnotenext
537 {\protected@xdef\@thefnmark{\thempfn}%
538 \@footnotetext}}%

```

(End of definition for `\footnotetext`.)

`\@xfootnotenext`

```

539 \def\@xfootnotenext[#1]{%
540 \begingroup
541 \csname c@\@mpfn\endcsname #1\relax
542 \unrestored@protected@xdef\@thefnmark{\thempfn}%
543 \endgroup
544 \@footnotetext}%

```

(End of definition for `\@xfootnotenext`.)

`\thempfn`

```

545 \def\@mpfn{footnote}
546 \def\thempfn{\thefootnote}%

```

(End of definition for `\thempfn` and `\@mpfn`.)

**\footref** This command generates a footnote mark. The value is produced by referencing a `\label` placed into a `\footnote` elsewhere (can be one in the main galley or in a minipage).

```
547 ⟨/2ekernel⟩
548 ⟨*2ekernel | latexrelease⟩
549 ⟨latexrelease⟩\IncludeInRelease{2021/06/01}%
550 ⟨latexrelease⟩ {\footref}{Add footref}%
551 \def\footref#1{%
552 \begingroup
553 \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
554 \endgroup
555 \footnotemark
556 }
557 ⟨/2ekernel | latexrelease⟩
558 ⟨latexrelease⟩\EndIncludeInRelease
```

We don't remove it when rolling back so that packages offered it in the past do not need to alter their behavior in a rollback situation.

```
559 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
560 ⟨latexrelease⟩ {\footref}{Add footref}%
561 ⟨latexrelease⟩
562 ⟨latexrelease⟩ % \let\footref\@undefined
563 ⟨latexrelease⟩
564 ⟨latexrelease⟩\EndIncludeInRelease
565 ⟨*2ekernel⟩
```

(*End of definition for \footref.*)

```
566 ⟨/2ekernel⟩
```

# File 46

## 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.
1 {*2ekernel}
2 \message{index,}
```

```

\makeindex

3 \def\makeindex{%
4 \newwrite\@indexfile
5 \immediate\openout\@indexfile=\jobname.idx
6 \def\index{\@bsphack\begingroup
7 \@sanitize
8 \wrindex}\typeout
9 {Writing index file \jobname.idx}%

```

Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```

10 \let\makeindex\empty
11 }
12 \onlypreamble\makeindex

```

*(End of definition for \makeindex.)*

```

\@wrindex

13 \def\@wrindex#1{%
14 \protected@write\@indexfile{}{%
15 \string\indexentry{#1}{\thepage}}%
16 \endgroup
17 \esphack}

```

*(End of definition for \@wrindex.)*

```

\index

18 \def\index{\@bsphack\begingroup \@sanitize\@index}

```

*(End of definition for \index.)*

```

\@index

19 \def\@index#1{\endgroup\esphack}

```

*(End of definition for \@index.)*

```

\makeglossary

20 \def\makeglossary{%
21 \newwrite\@glossaryfile
22 \immediate\openout\@glossaryfile=\jobname.glo
23 \def\glossary{\@bsphack\begingroup
24 \@sanitize
25 \wrglossary}\typeout
26 {Writing glossary file \jobname.glo }%

```

Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```

27 \let\makeglossary\empty
28 }
29 \onlypreamble\makeglossary

```

*(End of definition for \makeglossary.)*

```
\@wrglossary
30 \def\@wrglossary#1{%
31 \protected@write\@glossaryfile{}{%
32 {\string\glossaryentry{\#1}{\thepage}}%
33 \endgroup
34 \@esphack}
(End of definition for \@wrglossary.)
```

```
\glossary
35 \def\glossary{\@bsphack\begingroup\@sanitize\@index}
(End of definition for \glossary.)
36 </2ekernel>
```

## File 47

# 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{<file1,<file2, ...,<filen>}` : specifies the bibdata files. Writes a `\bibdata` entry on the `.aux` file and tries to read in `mainfile.bbl`.

`\bibliographystyle{<style>}` : Writes a `\bibstyle` entry on the `.aux` file.  
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{<citations>}` puts information on the `.aux` file that causes BIBTEX to include the `{<citations>}` list in the bibliography, but puts nothing in the text.

`\nocite{*}` is special: it tells BIBTEX to put the whole of a collection of references into the bibliography.

1 `(*2ekernel)`  
2 `\message{bibliography,}`

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

PARAMETERS

`\@cite` : A macro such that `\@cite{LABEL1,LABEL2}{NOTE}` produces the output for a `\cite[NOTE]{FOO1,FOO2}` command, where entry FOOi is defined by `\bibitem[LABELi]{FOOi}`.

The switch `@tempswa` is true if the optional NOTE argument is present.

The default definition is :

```
\@cite{LABELS}{NOTE} ==
BEGIN [LABELS
 IF @tempswa = T THEN , NOTE FI
]
END
```

`\@biblabel` : A macro to produce the label in the bibliography entry. For `\bibitem[LABEL]{NAME}`, the label is

generated by `\@biblabel{LABEL}`. It has the default definition `\@biblabel{LABEL} -> [LABEL]`.

## CONVENTION

`\b@FOO` : The name or number of the reference created by `\cite{FOO}`  
E.g., if `\cite{FOO} -> [17]`, then `\b@FOO -> 17`.

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```
\bibitem
 3 \def\bibitem{\@ifnextchar[\@lbibitem\@bibitem}
(End of definition for \bibitem.)

\@lbibitem
 4 \def@\lbibitem[#1]{\item[\@biblabel{#1}\hfill]\if@filesw
 5 {\let\protect\noexpand
 6 \immediate
 7 \write\auxout{\string\bibcite{#2}{#1}}}\fi\ignorespaces}
(End of definition for \@lbibitem.)

\@bibitem
 8 \def@\bibitem#1{\item\if@filesw \immediate\write\auxout
 9 {\string\bibcite{#1}{\the\value{\listctr}}}\fi\ignorespaces}
(End of definition for \@bibitem.)

\bibcite
10 \def\bibcite{\@newl@bel b}
(End of definition for \bibcite.)

\citation
11 \let\citation\@gobble
(End of definition for \citation.)

\cite
12 </2ekernel>
13 <*2ekernel | latexrelease>
14 <latexrelease>\IncludeInRelease{2022/06/01}%
15 <latexrelease> {\cite}{check for blank}%
16 \DeclareRobustCommand\cite{%
17 \@ifnextchar [{\@tempswattrue\@citex@checkblank}{\@tempswafalse\@citex@checkblank[]}}}
```

Due to the way `\for` as used in `\@citex` behaves an empty argument to `\cite` did not produce any warning for a missing citation. So we now inject a command before calling `\@citex` that does the checking for us. It is not done in `\@citex` directly, because that command is altered by a number of packages/classes and this way it is more likely that the check survives.

```
18 \def\@citex@checkblank[#1]{%
19 \IfBlankTF {#2}{%
20 {\@citex[#1]{\space}}{%
21 {\@citex[#1]{#2}}{%
22 }%
23 }%
24 }
```

```

24 ⟨latexrelease⟩\EndIncludeInRelease
25 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
26 ⟨latexrelease⟩ {\cite}{check for blank}%
27 ⟨latexrelease⟩
28 ⟨latexrelease⟩\DeclareRobustCommand\cite{%
29 ⟨latexrelease⟩ \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}}%
30 ⟨latexrelease⟩\let\@citex@checkblank\@undefined
31 ⟨latexrelease⟩
32 ⟨latexrelease⟩\EndIncludeInRelease
33 {*2ekernel}

```

(End of definition for `\cite`.)

`\@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)

```

34 \def\@citex[#1]{\leavevmode
35 \let\@citea\@empty
36 \@cite{\@for\@citereb:=#2\do
37 {\@citea\def\@citea{,\penalty\@m\ }%
38 \edef\@citereb{\expandafter\@firstofone\@citereb\@empty}%
39 \if@filesw\immediate\write\@auxout{\string\citation{\@citereb}}\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 Arseneau. 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.

```

40 \ifundefined{b@\@citereb}{\hbox{\reset@font\bfseries ?}}%
41 \G@refundefinedtrue
42 \@latex@warning
43 {Citation ‘\@citereb’ on page \thepage \space undefined}%
44 {\@cite@ofmt{\csname b@\@citereb\endcsname}}}\#1}}

```

(End of definition for `\@citex`.)

```

\bibdata
\bibstyle 45 \let\bibdata=\@gobble
46 \let\bibstyle=\@gobble

```

(End of definition for `\bibdata` and `\bibstyle`.)

```

\bibliography
47 \def\bibliography#1{%
48 \if@filesw
49 \immediate\write\@auxout{\string\bibdata{\zap@space#1 \@empty} }%
50 \fi
51 \input{\jobname.bbl}

```

(End of definition for \bibliography.)

```
\bibliographystyle
52 \def\bibliographystyle#1{%
53 \ifx\@begindocumenthook\@undefined\else
54 \expandafter\AtBeginDocument
55 \fi
56 {\if@filesw
57 \immediate\write\auxout{\string\bibstyle{#1}}%
58 \fi}}
```

(End of definition for \bibliographystyle.)

\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 \@citet, to get rid of leading spaces.

```
59 </2ekernel>
60 <*2ekernel | latexrelease>
61 <| latexrelease>\IncludeInRelease{2021/06/01}%
62 <| latexrelease> {\nocite}{Allow nocite in preamble}%
63 \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 $\varepsilon$  successor.]

Given that we are now a quarter century into using LATEX 2 $\varepsilon$  there is no good reason any more do limit ourself to 2.09 considerations. So we now simply delay the \nocite if it is issued in the preamble.

```
64 \ifx\@onlypreamble\document
```

Since we are after \begin{document} we can do the citations:

```
65 \@for\@citereb:=#1\do{%
66 \edef\@citereb{\expandafter\@firstofone\@citereb}%
67 \if@filesw\immediate\write\auxout{\string\citation{\@citereb}}\fi
68 \@ifundefined{b@\@citereb}{\G@refundefinedtrue
69 \G@warning{Citation ‘\@citereb’ undefined}}{}%
70 }
```

But before \begin{document} we raised an error message in the past but as of 2021/05 not any longer.

```
71 \% \@latex@error{Cannot be used in preamble}\@eha
```

Instead we delay the declaration to the start of the document. We have to use a late hook for this, so that it comes after the .aux file is open for writing and after \@preamblecmds was executed to change the above test. Therefore \AtBeginDocument would still be too early.

```
72 \AddToHook{begindocument/end}[kernel]{\nocite{#1}}%
73 \fi
```

```

74 \@esphack}
75 {/2ekernel | latexrelease}
76 \latexrelease\EndIncludeInRelease
77 \latexrelease\IncludeInRelease{0000/00/00}%
78 \latexrelease{}{\nocite}{Allow nocite in preamble}%
79 \latexrelease
80 \latexrelease\def\nocite#1{\@bsphack
81 \latexrelease\ifx\onlypreamble\document
82 \latexrelease\@for\@citeb:=#1\do{%
83 \latexrelease\edef\@citeb{\expandafter\firstofone\@citeb}%
84 \latexrelease\if@filesw\immediate\write\auxout{\string\citation{\@citeb}}\fi
85 \latexrelease\@ifundefined{b@\@citeb}{\G@refundefinedtrue
86 \latexrelease\@latex@warning{Citation ‘\@citeb’ undefined}}{}%
87 \latexrelease\else
88 \latexrelease\@latex@error{Cannot be used in preamble}\@eha
89 \latexrelease\fi
90 \latexrelease\@esphack}
91 \latexrelease
92 \latexrelease\EndIncludeInRelease
93 {/2ekernel}

```

Since `\nocite{*}` should not produce a warning about undefined citation keys (see PR 557), we need to set the control sequence ‘`\b@*`’ to something other than `\relax`. As a result `\cite{*}` will not warn either (but that never worked with BibTeX in the first place).

```
94 \expandafter\let\csname b@\endcsname\empty
```

(*End of definition for `\nocite`.*)

## 1.1 Default definitions

This hook determines the ‘relative formatting’ of the two logical parts of a citation with comment.

```
\@cite
95 \def\@cite#1#2{[#1\if@tempswa , #2\fi]}%
```

(*End of definition for `\@cite`.*)

`\@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.

```
96 \let\@cite@ofmt\hbox
```

(*End of definition for `\@cite@ofmt`.*)

```
\@biblabel
97 \def\@biblabel#1{[#1]}
98 {/2ekernel}
```

(*End of definition for `\@biblabel`.*)

# File 48

## lmarks.dtx

### Abstract

Marks are used to communicate information about the content of a page to the output routine. For example, in order to construct running headers, the output routine needs information about which section names are present on a page, and this information is passed to it through the mark system. However, marks may also be used for other purposes. This module provides a generalized mechanism for marks of independent classes.

## 1 Introduction

The *T<sub>E</sub>X* engines offer a low-level mark mechanism to communicate information about the content of the current page to the asynchronous operating output routine. It works by placing `\mark` commands into the source document. When the material for the current page is assembled in box 255, *T<sub>E</sub>X* scans for such marks and sets the commands `\topmark`, `\firstmark` and `\botmark`. The `\firstmark` receives the content of the first `\mark` seen in box 255 and `\botmark` the content of the last mark seen. The `\topmark` holds the content of the last mark seen on the previous page or more exactly the value of `\botmark` from the previous page. If there are no marks on the current page then all three are made equal to the `\botmark` from the previous page.

This mechanism works well for simple formats (such as plain *T<sub>E</sub>X*) whose output routines are only called to generate pages. It fails, however, in *L<sup>A</sup>T<sub>E</sub>X* (and other more complex formats), because here the output routine is sometimes called without producing a page, e.g., when encountering a float and placing it into one of the float regions. In that case the output routine is called, determines where to place the float, alters the goal for assembling text material (if the float was added to the top or bottom region) and then it resumes collecting textual material.

As a result the `\botmark` gets updated and so `\topmark` no longer reflects the situation at the top of the next page when that page is finally boxed.

Another problem for *L<sup>A</sup>T<sub>E</sub>X* was that it wanted to use several “independent” marks and in the early implementations of *T<sub>E</sub>X* there was only a single `\mark` command available. For that reason *L<sup>A</sup>T<sub>E</sub>X* implemented its own mark mechanism where the marks always contained two parts with their own interfaces: `\markboth` and `\markright` to set marks and `\leftmark` and `\rightmark` to retrieve them.

However, this extended mechanism (while supporting scenarios such as chapter/section marks) was far from general. The mark situation at the top of a page (i.e., `\topmark`) remained unusable and the two marks offered were not really independent of each other because `\markboth` (as the name indicates) was always setting both.

The new mechanism overcomes both issues:

- It provides arbitrarily many, fully independent named marks, that can be allocated and, from that point onwards, used.
- It offers access for each such marks to retrieve its top, first, and bottom values separately.
- Furthermore, the mechanism is augmented to give access to marks in different “regions” which may not be just full pages.

## 2 Design-level and code-level interfaces

The interfaces are mainly meant for package developers, but they are usable (with appropriate care) also in the document preamble, for example, when setting up special running headers with `fancyhdr`, etc. They are therefore available both as CamelCase commands as well as commands for use in the L3 programming layer. Both are described together below.

---

```
\NewMarkClass \NewMarkClass {<class>}
\mark_new_class:n \mark_new_class:n {<class>}
```

---

Declares a new `<class>` of marks to be tracked by L<sup>A</sup>T<sub>E</sub>X. Each `<class>` must be declared before it is used.

Mark classes can only be declared before `\begin{document}`.

---

```
\InsertMark \InsertMark {<class>} {<text>}
\mark_insert:nn \mark_insert:nn {<class>} {<text>}
```

---

Adds a mark to the current galley for the `<class>`, containing the `<text>`.

It has no effect in places in which you can't place floats, e.g., a mark inside a box or inside a footnote never shows up anywhere.

If used in vertical mode it obeys L<sup>A</sup>T<sub>E</sub>X's internal `\nobreak` switch, i.e., it does not introduce a breakpoint if used after a heading. If used in horizontal mode it doesn't handle spacing (like, for example, `\index` or `\label` does, so it should be attached to material that is typeset).

---

```
insertmark \AddToHook {insertmark} {<code>}
```

---

When marks are inserted, the mark content may need some special treatment, e.g., by default `\label`, `\index`, and `\glossary` do not expand at this time (but only later if and when the mark content is actually used). In order to allow packages to augment or alter this setup there is a public hook `insertmark` that is executed at this point. It runs in a group so local modification to commands are only applied to the `<text>` argument of `\InsertMark` or `\mark_insert:nn`.

---

```

\TopMark * \TopMark [{<region>}]{<class>}
\FirstMark * \FirstMark [{<region>}]{<class>}
\LastMark * \LastMark [{<region>}]{<class>}
\mark_use_top:nn * \mark_use_top:nn [{<region>}]{<class>}
\mark_use_first:nn * \mark_use_first:nn [{<region>}]{<class>}
\mark_use_last:nn * \mark_use_last:nn [{<region>}]{<class>}

```

---

These functions expand to the appropriate mark  $\langle text \rangle$  for the given  $\langle class \rangle$  in the specified  $\langle region \rangle$ . The default  $\langle region \rangle$  in the design-level commands is `page`. Note that with the L3 layer commands there are no optional arguments, i.e., both arguments have to be provided.

**TeXhackers note:** The result is returned within the `\unexpanded` primitive (`\exp_not:n`), which means that the  $\langle text \rangle$  does not expand further when appearing in an `x`-type or `e`-type argument expansion.

The “first” and “last” marks are those seen first and last in the current region/page, respectively. The “top” mark is the last mark of the  $\langle class \rangle$  seen in an earlier region, i.e., the  $\langle text \rangle$  what would be “current” at the very top of the region.

### Important!

The commands are only meaningful inside the output routine, in other places their result is (while not random) unpredictable due to the way `LATeX` cuts text material into pages.

Currently,  $\langle region \rangle$  is one of `page`, `previous-page`, `column`, and `previous-column`. If a page has just been finished then the region `page` refers to the current page and `previous-page`, as the name indicates, to the page that has been finished previously. This means you are able to access mark information for the current page as well as for the page before if you are inside the output routine, without the need to explicitly save that information beforehand.

In single column documents the `column` is the same as the `page` region, but in two-column documents, `column` refers to the current column that just got finished and `previous-column` to the one previously finished. Code for running headers are (in standard `LATeX`) only evaluated when both columns are assembled, which is another way of saying that in that case `previous-column` refers to the left column and `column` to the right column. However, to make this a bit nicer to access, there are also alias regions named `first-column` and `last-column`<sup>40</sup> to access these regions.<sup>41</sup>

Note that you can only look backwards at already processed regions, e.g., in a `twoside` document finishing a recto (odd, right-hand) page you can access the data from the facing verso (left-hand) page, but if you are finishing a left-hand page you can't integrate data from the upcoming right-hand page. If such a scenario needs to be realized then it is necessary to save the left-hand page temporarily instead of finalizing it, process material for the right-hand page and once both are ready, attach running headers and footers and shipout out both in one go.<sup>42</sup>

<sup>40</sup>This is called “last” not “second” in anticipation of extending the mechanism to multiple columns, where first and last would still make sense.

<sup>41</sup>At the moment there aren't any `previous-...-column` regions to access the columns from the previous page. If necessary, the mechanism could be easily augmented to cover them too, though.

<sup>42</sup>As of now that scenario is not yet officially supported.

---

```
\IfMarksEqualTF * \IfMarksEqualTF [<region>] {<class>} {<pos1>} {<pos2>} {<true>} {<false>}
\IfMarksEqualT * \mark_if_eq:nnnnTF [<region>] {<class>} {<pos1>} {<pos2>} {<true>} {<false>}
\IfMarksEqualF * \mark_if_eq:nnnnnTF [<region1>] {<class1>} {<pos1>}
 {<region2>} {<class2>} {<pos2>} {<true>} {<false>}
\mark_if_eq:nnnnTF *
\mark_if_eq:nnnnnnTF *
```

These conditionals allow you to compare the content of two marks and act based on the result. The commands work in an expansion context, if necessary.

It is quite common when programming with marks to need to interrogate conditions such as whether marks have appeared on a previous page, or if there are multiple marks present on the current page, and so on. The tests above allow for the construction of a variety of typical test scenarios, with three examples presented below.

The first two conditionals cover only the common scenarios. Both marks are picked up from the same `<region>` (by default `page`) and they have to be of the same `<class>`.<sup>43</sup> The `<posi>` argument can be either `top`, `first`, or `last`.

Important to note is that the comparison is not with respect to the textual content of the marks but whether or not they originated from the same `\InsertMark` command (or the L3 layer version `\mark_insert:nn`).

If you wish to compare marks across different regions or across different classes, you have to do it using the generic test only available in the L3 programming layer or do it manually, i.e., get the marks and then compare the values yourself.<sup>44</sup>

## 2.1 Use cases for conditionals

However, the basic version is enough for the following typical use cases:

**Test for at most one mark of class myclass on current page:** If the first and last mark in a region are the same then either there was no mark at all, or there was at most one. To test this on the current page:

```
\NewMarkClass{myclass}
\IfMarksEqualTF{myclass}{first}{last}
 { <zero or one mark> }{ <two or more marks> }
```

**Test for no mark of class myclass in the previous page:** If the top mark is the same as the first mark, there is no mark in the region at all. If we wanted to do this test for the previous page:

```
\IfMarksEqualTF[previous-page]{myclass}{top}{first}
 { <no marks> }{ <at least one mark> }
```

Comparing `top` and `last` would give you the same result.

**Test for zero, one, or more than one:** Combining the two tests from above you can test for zero, one or more than one mark.

```
\IfMarksEqualTF{myclass}{top}{first}
 { <no marks> }
 {\IfMarksEqualTF{myclass}{first}{last}
 { <exactly one mark> }{ <more than one mark> }}
```

<sup>43</sup>If an undeclared mark class is used the tests return `true` (not an error).

<sup>44</sup>If two undeclared mark classes are compared the result is always `true`; if a declared and an undeclared mark class is used it is always `false`.

If you need one of such tests more often (or if you want a separate command for it for readability), then consider defining:

```
\providecommand\IfNoMarkTF[2][page]{\IfMarksEqualTF[#1]{#2}{first}{last}}
```

## 2.2 Debugging mark code

---

```
\DebugMarksOn \DebugMarksOn ... \DebugMarksOff
\DebugMarksOff
\mark_debug_on:
\mark_debug_off:
```

Commands to turn the debugging of mark code on or off. The debugging output is rather coarse and not really intended for normal use at this point in time.

## 3 Application examples

If you want to figure out if a break was taken at a specific point, e.g., whether a heading appears at the top of the page, you can do something like this:

```
\newcounter{breakcounter}
\NewMarkClass{break}
\newcommand\markedbreak[1]{\stepcounter{breakcounter} %
 \InsertMark{break}{\arabic{breakcounter}}%
 \penalty #1\relax
 \InsertMark{break}{-\arabic{breakcounter}}}
```

To test if the break was taken you can test if `\TopMark{break}` is positive (taken) or negative (not taken) or zero (there was never any marked break so far). The absolute value can be used to keep track of which break it was (with some further coding).

*to be extended with additional application examples*

## 4 Legacy L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> interface

Here we describe the interfaces that L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  offered since the early nineties and some minor extensions.

### 4.1 Legacy design-level and document-level interfaces

---

```
\markboth \markboth {\left} {\right}
\markright \markright {\right}
```

L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  uses two marks which aren't fully independent. A "left" mark generated by the first argument of `\markboth` and a "right" mark generated by the second argument of `\markboth` or by the only argument of `\markright`. The command `\markboth` and `\markright` are in turn called from heading commands such as `\chaptermark` or `\sectionmark` and their behavior is controlled by the document class.

For example, in the `article` class with `twoside` in force the `\sectionmark` will issue `\markboth` with an empty second argument and `\subsectionmark` will issue `\markright`. As a result the left mark will contain chapter titles and the right mark subsection titles.

Note, however, that in one-sided documents the standard behavior is that only `\markright` is used, i.e., there will only be right-marks but no left marks!

---

```
\leftmark * \leftmark
\rightmark * \rightmark
```

---

These functions return the appropriate mark value from the current page and work as before, that is `\leftmark` will get the last (!) left mark from the page and `\rightmark` the first (!) right mark.

In other words they work reasonably well if you want to show the section title that is current when you are about to turn the page and also show the first subsection title on the current page (or the last from the previous page if there wasn't one). Other combinations can't be shown using this interface.

The commands are fully expandable, because this is how they have been always defined in L<sup>A</sup>T<sub>E</sub>X. However, this is of course only true if the content of the mark they return is itself expandable and does not contain any fragile material. Given that this can't be guaranteed for arbitrary content, a programmer using them in this way should use `\protected@edef` and *not* `\edef` to avoid bad surprises as far as this is possible, or use the new interfaces (`\TopMark`, `\FirstMark`, and `\LastMark`) which return the `\text` in `\exp_not:n` to prevent uncontrolled expansion.

## 4.2 Legacy interface extensions

The new implementation adds three mark classes: `2e-left`, `2e-right` and `2e-right-nonempty` and patches `\markboth` and `\markright` slightly so that they also update these new mark classes, so that the new classes work with existing document classes.

As a result you can use `\LastMark{2e-left}` and `\FirstMark{2e-right}` instead of `\leftmark` and `\rightmark`. But more importantly, you can use any of the other retrieval commands to get a different status value from those marks, e.g., `\LastMark{2e-right}` would return the last subsection on the page (instead of the first as returned by `\rightmark`).

The difference between `2e-right` and `2e-right-nonempty` is that the latter will only be updated if the material for the mark is not empty. Thus `\markboth{title}{}{}` as issued by, say, `\sectionmark`, sets a `2e-left` mark with `title` and a `2e-right` mark with the empty string but does not add a `2e-right-nonempty` mark.

Thus, if you have a section at the start of a page and you would ask for `\FirstMark{2e-right}` you would get an empty string even if there are subsections on that page. But `2e-right-nonempty` would then give you the first or last subsection on that page. Of course, nothing is simple. If there are no subsections it would tell you the last subsection from an earlier page. We therefore need comparison tools, e.g., if top and first are identical you know that the value is bogus, i.e., a suitable implementation would be

```
\IfMarksEqualTF{2e-right-nonempty}{top}{first}
 { <appropriate action if there was no real mark> }
 {\FirstMark{2e-right-nonempty}}
```

## 5 Notes on the mechanism

In contrast to vanilla T<sub>E</sub>X,  $\varepsilon$ -T<sub>E</sub>X extends the mark system to allow multiple independent marks. However, it does not solve the `\topmark` problem which means that L<sup>A</sup>T<sub>E</sub>X still needs to manage marks almost independently of T<sub>E</sub>X. The reason for this is that the more complex output routine used by L<sup>A</sup>T<sub>E</sub>X to handle floats (and related structures)

means that `\topmark(s)` remain unreliable. Each time the output routine is fired up, `TEX` moves `\botmark` to `\topmark`, and while  $\varepsilon$ -`TEX` extends this to multiple registers the fundamental concept remains the same. That means that the state of marks needs to be tracked by `LATEX` itself. An early implementation of this package used `TEX`'s `\botmark` only to ensure the correct interaction with the output routine (this was before the  $\varepsilon$ -`TEX` mechanism was even available). However, other than in a prototype implementation for `LATEX3`, this package was never made public.

The new implementation now uses  $\varepsilon$ -`TEX`'s marks as they have some advantages, because with them we can leave the mark text within the galley and only extract the marks during the output routine when we are finally shipping out a page or storing away a column for use in the next page. That means we do not have to maintain a global data structure that we have to keep in sync with informational marks in the galley but can rely on everything being in one place and thus manipulations (e.g. reordering of material) will take the marks with them without a need for updating a fragile linkage.

To allow for completely independent marks we use the following procedure:

- For every type of marks we allocate a mark class so that in the output routine `TEX` can calculate for each class the current top, first, and bottom mark independently. For this we use `\newmarks`, i.e., one marks register per class.
- As already mentioned firing up an output routine without shipping out a page means that `TEX`'s top marks get wrong so it is impossible to rely on `TEX`'s approach directly. What we do instead is to keep track of the real marks (for the last page or more generally last region) in some global variables.
- These variables are updated in the output routine at defined places, i.e., when we do real output processing but not if we use special output routines to do internal housekeeping.
- The trick we use to get correctly updated variables is the following: the material that contains new marks (for example the page to be shipped out) is stored in a box. We then use `TEX` primitive box splitting functions by splitting off the largest amount possible (which should be the whole box if nothing goes really wrong). While that seems a rather pointless thing to do, it has one important side effect: `TEX` sets up first and bottom marks for each mark class from the material it has split off. This way we get the first and last marks (if there have been any) from the material in the box.
- The top marks are simply the last marks from the previous page or region. And if there hasn't been a first or bottom mark in the box then the new top mark also becomes new first and last mark for that class.
- That mark data is then stored in global token lists for use during the output routine and legacy commands such as `\leftmark` or new commands such as `\TopMark` simply access the data stored in these token lists.

That's about it in a nutshell. Of course, there are some details to be taken care of—those are discussed in the implementation sections.

## 6 Internal output routine functions

The functions in this section are tied to the output routine and used in the interface to L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> and perhaps at some later time within a new output routine for L<sup>A</sup>T<sub>E</sub>X. They are not meant for general use and are therefore made internal. Internal means that @<sub>0</sub> automatically gets replaced in the code (and in the documentation) so we have to give it a suitable value.

```
1 (@@=mark)
```

---

### \\_\\_mark\\_update\\_singlecol\\_structures: \\_\\_mark\\_update\\_singlecol\\_structures:

L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> integration function in case we are doing single column layouts. It assumes that the page content is already stored in \@outputbox and processes the marks inside that box. It is called as part of \@opcol.

---

### \\_\\_mark\\_update\\_dblcol\\_structures: \\_\\_mark\\_update\\_singlecol\\_structures:

L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> integration function mark used when we are doing double column documents. It assumes that the page content is already stored in \@outputbox and processes the marks inside that box. It then does different post-processing depending on the start of the switch \if@firstcolumn. If we are in the second column it also has to update page marks, otherwise it only updates column marks. It too is called as part of \@opcol.

---

### \\_\\_mark\\_update\\_structure\\_from\\_material:nn \\_\\_mark\\_update\\_structure\\_from\\_material:nn {\<region>} {\<material with marks>}

Helper function that inspects the marks inside the second argument and assigns new mark values based on that to the <region> given in the first argument. For this it first copies the mark structure from <region> to previous-<region> and then takes all last mark values currently in the region and makes them the new top mark values. Finally it assigns new first and last values for all mark classes based on what was found in the second argument.

As a consequence, the allowed values for <region> are `page` and `column` because only they have `previous-...` counterparts.

Another important aspect to keep in mind is that marks are recognized only if they appear on the top level, e.g., if we want to process material stored in boxes we need to put it unboxed (using \unvcopy etc.) into the second argument.

---

### \\_\\_mark\\_update\\_structure\\_alias:nn \\_\\_mark\\_update\\_structure\\_alias:nn {\<alias>} {\<source>}

Helper function that copies all mark values in the <source> region to <alias>, i.e., make the structures identical. Used to update the `previous-...` structures inside \\_\\_mark\\_update\\_structure\\_from\\_material:nn and `first-column` and `last-column` structures inside \\_\\_mark\\_update\\_singlecol\\_structures: or \\_\\_mark\\_update\\_dblcol\\_structures:.

---

### \\_\\_mark\\_update\\_structure\\_to\\_err:n \\_\\_mark\\_update\\_structure\\_to\\_err:n {\<region>}

Helper function that sets all mark values in the <region> to an error message. This is currently used for `last-column` at times where using marks from it would be questionable/wrong, i.e., when we have just processed the first column in a two-column document.

---

```
__mark_get_marks_for_reinsertion:nNN __mark_get_marks_for_reinsertion:nNN {<source>}
 ⟨token-list-var for collecting first marks⟩
 ⟨token-list-var for collecting last marks⟩
```

---

Helper function for extracting marks that would otherwise get lost, for example when they are hidden inside a box. This helper does not update mark structures and can therefore be used outside the output routine as well.

It collects all the top-level marks from inside the `<source>` and adds suitable `\mark_insert:nn` in the two token lists. These token lists can then be executed at the right place to reinsert the marks, e.g., directly after the box. This is, for example, going to be used<sup>45</sup> by `multicol` when a short balanced `multicols` is returned to the galley for typesetting.

If the `<source>` consists of a single vertical box (plus possibly followed by some glue but nothing else) then the box is unpacked and the top-level marks are collected from its content. However, if it is not a vertical box or there are other data then nothing is unpacked and you have to do the unpacking yourself to get at the marks inside.

It is quite likely that one only needs a single token list for returning the `\mark_insert:nn` statements. If that is the case this command may change to take only two arguments.

## 7 The Implementation

```
2 {*2ekernel | latexrelease}
3 \ExplSyntaxOn
4 ⟨latexrelease⟩\NewModuleRelease[2022/06/01]{ltmarks}
5 ⟨latexrelease⟩ {Marks~handling}
```

### 7.1 Allocating new mark classes

---

`\g_mark_classes_seq` A list holding all the mark classes that have been declared.

```
6 \seq_new:N \g_mark_classes_seq
```

`\mark_new_class:n` `\__mark_new_class:nn` A mark class is created by initializing a number of data structures. First, we get a register number to refer to the mark class. The new mark class is then added to the `\g_mark_classes_seq` sequence to be able to easily loop over all classes. Finally a number of top-level global token lists are declared that hold various versions of the mark for access.

```
7 \cs_new_protected:Npn \mark_new_class:n #1
8 {
9 \seq_if_in:NnTF \g_mark_classes_seq {#1}
10 {
11 \msg_error:nnn { mark } { class-already-defined }
12 {#1}
13 }
14 { __mark_new_class:nn {#1} }
15 }
```

This is only available in the preamble.

```
16 \onlypreamble \mark_new_class:n
```

---

<sup>45</sup>Probably not before 2025, though.

The internal command carries out the necessary allocations.

```

17 \cs_new_protected:Npn __mark_new_class:nn #1
18 {
19 (*trace)
20 __mark_debug:n { \iow_term:x { Marks:-new-mark:-#1-\msg_line_context: } }
21 (/trace)

```

Use the L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\epsilon$</sub>  interface for now as the L3 programming layer doesn't have one for marks yet.

```
22 \exp_args:Nc \newmarks {c_mark_class_ #1 _mark}
```

Remember the new class in the sequence.

```
23 \seq_gput_right:Nn \g_mark_classes_seq {#1}
```

We need three token lists for each region, one for top, first, and last.

```

24 \tl_new:c { g_mark_page_top_ #1 _tl }
25 \tl_new:c { g_mark_page_first_ #1 _tl }
26 \tl_new:c { g_mark_page_last_ #1 _tl }

```

For the page region we also keep track of the previous-page.

```

27 \tl_new:c { g_mark_previous-page_top_ #1 _tl }
28 \tl_new:c { g_mark_previous-page_first_ #1 _tl }
29 \tl_new:c { g_mark_previous-page_last_ #1 _tl }

```

Same game for column and previous-column

```

30 \tl_new:c { g_mark_column_top_ #1 _tl }
31 \tl_new:c { g_mark_column_first_ #1 _tl }
32 \tl_new:c { g_mark_column_last_ #1 _tl }
33 \tl_new:c { g_mark_previous-column_top_ #1 _tl }
34 \tl_new:c { g_mark_previous-column_first_ #1 _tl }
35 \tl_new:c { g_mark_previous-column_last_ #1 _tl }

```

But for columns we also allocate token lists for the alias regions `first-column` and `last-column`.

```

36 \tl_new:c { g_mark_first-column_top_ #1 _tl }
37 \tl_new:c { g_mark_first-column_first_ #1 _tl }
38 \tl_new:c { g_mark_first-column_last_ #1 _tl }
39 \tl_new:c { g_mark_last-column_top_ #1 _tl }
40 \tl_new:c { g_mark_last-column_first_ #1 _tl }
41 \tl_new:c { g_mark_last-column_last_ #1 _tl }

```

All marks will have an identification at the beginning of the form `\__mark_id:n{<number>}` and therefore the initial empty values should have that too, so that data extraction is going to be uniform.

```

42 \tl_set:cn { g_mark_page_top_ #1 _tl }{ __mark_id:n{0} }
43 \tl_set:cn { g_mark_page_first_ #1 _tl }{ __mark_id:n{0} }
44 \tl_set:cn { g_mark_page_last_ #1 _tl }{ __mark_id:n{0} }
45 \tl_set:cn { g_mark_previous-page_top_ #1 _tl }{ __mark_id:n{0} }
46 \tl_set:cn { g_mark_previous-page_first_ #1 _tl }{ __mark_id:n{0} }
47 \tl_set:cn { g_mark_previous-page_last_ #1 _tl }{ __mark_id:n{0} }
48 \tl_set:cn { g_mark_column_top_ #1 _tl }{ __mark_id:n{0} }
49 \tl_set:cn { g_mark_column_first_ #1 _tl }{ __mark_id:n{0} }
50 \tl_set:cn { g_mark_column_last_ #1 _tl }{ __mark_id:n{0} }
51 \tl_set:cn { g_mark_previous-column_top_ #1 _tl }{ __mark_id:n{0} }
52 \tl_set:cn { g_mark_previous-column_first_ #1 _tl }{ __mark_id:n{0} }
53 \tl_set:cn { g_mark_previous-column_last_ #1 _tl }{ __mark_id:n{0} }

```

```

54 \tl_set:cn { g_mark_first-column_top_ #1 _tl }{ __mark_id:n{0} }
55 \tl_set:cn { g_mark_first-column_first_ #1 _tl }{ __mark_id:n{0} }
56 \tl_set:cn { g_mark_first-column_last_ #1 _tl }{ __mark_id:n{0} }
57 \tl_set:cn { g_mark_last-column_top_ #1 _tl }{ __mark_id:n{0} }
58 \tl_set:cn { g_mark_last-column_first_ #1 _tl }{ __mark_id:n{0} }
59 \tl_set:cn { g_mark_last-column_last_ #1 _tl }{ __mark_id:n{0} }
60 }

```

(End of definition for `\mark_new_class:n` and `\__mark_new_class:nn`. This function is documented on page 943.)

## 7.2 Updating mark structures

```

\l_mark_box
\l_mark_ii_box
\g_mark_tmp_tl
\g_mark_new_top_tl

```

For some operations we need two temporary private boxes and two private global token lists.

```

61 \box_new:N \l_mark_box
62 \box_new:N \l_mark_ii_box
63 \tl_new:N \g_mark_tmp_tl
64 \tl_new:N \g_mark_new_top_tl

```

(End of definition for `\l_mark_box` and others.)

```

__mark_extract_and_handle_marks:nn

```

This is the main macro to extract and handle marks inside some vertical material. It is used by `\__mark_update_structure_from_material:nn` (for updating the mark structure for a region based on the marks found) and by `\__mark_get_marks_for_reinsertion:nnn` (for extracting marks from some material and prepare for reinserting them later (e.g., out of a box that is placed as a box into the main galley)).

```

65 \cs_new_protected:Npn __mark_extract_and_handle_marks:nn #1#2 {

```

This macro expects code to handle extracted marks in its first argument and vertical material (not boxed or just consisting of a single vertical box) as its second. It extracts top-level mark information from #2, stores them as split marks and then calls #1 to make use of this information.

If it finds a forced break in the material it removes it and then restarts the attempt without it.

We start with a group to keep most changes local.

```

66 \group_begin:

```

Getting the first and last marks out of the material in #2 is done by putting the material in a box and then doing a split operation to the maximum size possible (which hopefully gets us all of the content).<sup>46</sup> Because this action is used only to get the mark values, we don't want any underfull box warnings so we (locally) turn those off.

```

67 \dim_set_eq:NN \tex_splitmaxdepth:D \c_max_dim
68 \int_set_eq:NN \tex_vbadness:D \c_max_int
69 \dim_set_eq:NN \tex_vfuzz:D \c_max_dim

```

There is a further complication: if the material contains infinite shrinking glue then a `\vsplit` operation will balk with a low-level error. Now pages or columns, which are our main concern here, can't have such infinite shrinkage if they are cut straight from the galley, however the use of `\enlargethispage` actually does add some at the very bottom

---

<sup>46</sup>With normal column material cut from the main galley we should always get all material in one go, but in certain situations, for example, in a `multicols` environment that contains some `\columnbreaks` a single split operation will not be enough. Thus, this is something we need to handle.

(and also wraps the whole page into a box by itself, so if we leave it this way then a) we get this error and b) we don't see any marks because they are hidden one level down).

Another possible issue are packages or user code that place stray `\vboxes` directly into the main galley (an example is `marginnote` that attaches its marginals in this way). If such boxes end up as the last item on the page we should not unpack them.

All these issues need to be handled, which is done in `\_mark_prepare_and_extract:nn`.

```
70 _mark_prepare_and_extract:nn {#1} {#2}
```

Once all mark classes have been processed, the data structures are updated and we can close the group, which undoes our local changes and retains only the global ones.

```
71 \group_end:
72 }
```

*(End of definition for \\_mark\_extract\_and\_handle\_marks:nn.)*

`\_mark_prepare_and_extract:nn` This macro does the dirty work. It is not directly integrated in `\_mark_extract_and_handle_marks:nn` because we may have to call it recursively if we find forced breaks.

```
73 \cs_new_protected:Npn _mark_prepare_and_extract:nn #1#2 {
```

To handle the `\enlargethispage` case we do an `\unskip` to get rid of any glue that is present at the very end of the material and also check if we have then a `\vbox` as the last item and if so unpack that too, but only under certain conditions, see below. All this is temporary done in a group, just for getting the marks out, so it doesn't affect the final page production.

```
74 \vbox_set:Nn \l__mark_box
75 {
76 #2
77 \tex_unskip:D
78 \box_set_to_last:N \l__mark_box
```

After having removed the last box from the current list (if there was one) we check whether the vertical list is now empty. If not, then the last box is definitely not the one from `\enlargethispage` and so we can, and should, leave it alone. Otherwise we check if this last box is a `\vbox`.

```
79 \int_compare:nNnT \tex_lastnodetype:D < 0
80 {
81 \box_if_vertical:NT \l__mark_box
```

If it is, we unpack the box.

```
82 { \vbox_unpack:N \l__mark_box }
83 }
```

If it wasn't a `vbox`, it was either an `hbox` or there was no box. Given that we are only interested in the marks we don't need put it back in that case.

```
84 }
```

We are now ready to `\vsplit` the box to get at the marks. If the box contains some infinite negative glue the `\TeX` will produce an error complaining about it but it will correctly find the the split marks. Given that we can't prevent that error, we hide it from the user and ensure that `\TeX` doesn't stop. The error message still shows in the log, but even that is mitigated as best as possible—see the definition of `\_mark_vbox_set_split_to_maxdimen:NN` for the tricks employed.

```
85 _mark_vbox_set_split_to_maxdimen:NN \l__mark_ii_box \l__mark_box
```

After splitting we check if there is anything left in `\l__mark_box`. If not then the above split has set some split marks that we can then use to finish the extraction:

```
86 \box_if_empty:NTF \l__mark_box
87 { #1 }
```

If we have a remainder after the split then this means that there was some forced break in the material. We get rid of that by combining the content of the two boxes and restart.

```
88 {
89 (*trace)
90 __mark_debug:n { \iow_term:x
91 { Marks:~ mark~ extraction-needs~ recursion~
92 \msg_line_context: } }
93 (/trace)
94 __mark_prepare_and_extract:nn {#1}
95 { \vbox_unpack:N \l__mark_ii_box
96 \vbox_unpack:N \l__mark_box }
97 }
98 }
```

*(End of definition for `\__mark_prepare_and_extract:nn`.)*

`\__mark_vbox_set_split_to_maxdimen:NN` Split a box to get at its marks without pausing even if TeX is producing an error message because of infinite negative glue in the box. If there is such an error we ensure that it only shows up in the log but not on the terminal.

The nice low-level hack by DPC records in the .log that a glue shrinkage error is harmless.

We disguise `\c_max_dim` in an odd looking csname, which then shows up as part of the display of an error message if that error happens. This csname forms part of the error display so what you get is something like

```
! Infinite glue shrinkage found in box being split.
<argument> Infinite shrink error above ignored !
1. ... }
```

which hopefully makes it clear that the error is harmless and and should be ignored by the reader of the .log.

```
99 \cs_set_eq:cN {Infinite~shrink~error~above~ignored-!}\c_max_dim
```

The whole definition of `\__mark_vbox_set_split_to_maxdimen:NN` below is fully expanded, so we have to use a lot of `\exp_not:N` commands to prevent expansion where necessary.

```
100 \cs_new_protected:Npx __mark_vbox_set_split_to_maxdimen:NN #1#2 {
```

We start by saving the current interaction and escape char settings.

```
101 \tl_set:Ne \exp_not:N \l__mark_saved_parameters_tl
102 {
103 \tex_interactionmode:D
104 \exp_not:N \int_use:N \tex_interactionmode:D \scan_stop:
105 \tex_escapechar:D
106 \exp_not:N \int_use:N \tex_escapechar:D \scan_stop:
107 }
```

Then we change them so that no escape char is printed in the error message (accounts for the missing backslash in front of `Infinite shrink ...`) and we set the interaction to `\nonstopmode` so that the the error (if any) just goes into the `.log` file and TeX doesn't stop at that point.

```
108 \tex_escapechar:D -1 \scan_stop:
109 \tex_interactionmode:D 0 \scan_stop:
```

Then we do the splitting of the box to `\c_max_dim` to get at the marks. This may generate the error we are worried about, i.e., if the box contains infinite negative glue. However, TeX makes this glue finite and continues, which means we get our split marks which is really all we care about.

```
110 \tex_setbox:D #1 \tex_vsplit:D #2 to
```

The `\use:n` may seem pointless, and it is to some extent, but we need it to get our disguised `\c_max_dim` displayed properly as part of the error message if there is one. Without it, the display would show only part of what we want it to show (try it).

```
111 \exp_not:N \use:n {
112 \use:c{Infinite~shrink~error~above~ignored-!}
113 }
```

Finally, we change the escape char and the interaction mode back to what it was before:

```
114 \exp_not:N \l__mark_saved_parameters_tl
115 }
```

*(End of definition for `\__mark_vbox_set_split_to_maxdimen:NN`.)*

`\l__mark_saved_parameters_tl` The temporary variable used for resetting escape char and interaction mode.

```
116 \tl_new:N \l__mark_saved_parameters_tl
```

*(End of definition for `\l__mark_saved_parameters_tl`.)*

`\__mark_update_structure_from_material:nn` This function updates the mark structures of a region. The first argument is the region to update and second argument receives the material that holds the marks. Out of this material we extract the first and last marks for all classes (if there are any) to do the assignments.

```
117 \cs_new_protected:Npn __mark_update_structure_from_material:nn #1#2 {
118 __mark_extract_and_handle_marks:nn
```

Once the marks can be extracted we update the structure from the split marks (code in `\__mark_update_structure_from_splitmarks:n`).

```
119 { __mark_update_structure_from_splitmarks:n {#1} }
120 { #2 }
121 }
```

*(End of definition for `\__mark_update_structure_from_material:nn`.)*

`\__mark_update_structure_from_splitmarks:n` This macro is called after we have done a `\tex_vsplit:D` operation and the mark data is in the split marks.

```
122 \cs_new_protected:Npn __mark_update_structure_from_splitmarks:n #1 {
```

The first thing we do is to copy the current region structure to `previous-...`; this leaves the current structure untouched so we can update it class by class (which is necessary).

```
123 __mark_update_structure_alias:nn { previous-#1 } {#1}
```

After this action we can get first and last marks of the various classes through `\tex_splitfirstmarks:D` and `\tex_splitbotmarks:D`. So now we loop over all classes stored in `\g_mark_classes_seq`.

```
124 \seq_map_inline:Nn \g_mark_classes_seq
125 {
```

First action: get the last mark from the previous region, i.e., `previous-#1`. But because it is also still inside `#1`, at the moment we use that to construct the name because this is a tiny bit faster. Given that we need this value in various assignments we store it away which avoids unnecessary further csname generations.

```
126 \tl_gset_eq:Nc \g_mark_new_top_tl { g_mark_#1_last_##1_t1 }
```

This will first of all become the new top mark for the current class.

```
127 \tl_gset_eq:cN { g_mark_#1_top_##1_t1 } \g_mark_new_top_tl
```

Next action is to get ourselves the new last mark from the material supplied.

```
128 \tl_gset:No \g_mark_tmp_t1
129 { \tex_splitbotmarks:D \use:c { c_mark_class_##1_mark } }
```

If this mark doesn't exist then obviously neither does the first mark, so both become the last mark from the previous region. We have to be a little careful here: something like `\mark_insert:nn{foo}{}{}` adds an “empty” mark that should not be confused with no mark at all. But no mark in our material will result in `\g_mark_tmp_t1` being fully empty. This is why we have to make sure that “empty” from `\mark_insert:nn` only appears to be empty when typeset but fails the next test (see below how this is done).

```
130 \tl_if_empty:NTF \g_mark_tmp_t1
131 {
132 \tl_gset_eq:cN { g_mark_#1_last_##1_t1 }
133 \g_mark_new_top_t1
134 \tl_gset_eq:cN { g_mark_#1_first_##1_t1 }
135 \g_mark_new_top_t1
136 }
```

If it wasn't empty, i.e., if it had a real value then we use this value for our new last mark instead.

```
137 {
138 \tl_gset_eq:cN { g_mark_#1_last_##1_t1 } \g_mark_tmp_t1
```

Because we had a last mark we also have a first mark (which might be the same, but might be not), so we pick that up and assign it to the appropriate token list. This explains why we first checked for the last mark because that makes the processing faster in case there is none.

```
139 \tl_gset:co { g_mark_#1_first_##1_t1 }
140 {
141 \tex_splitfirstmarks:D
142 \use:c { c_mark_class_##1_mark }
143 }
144 }
145 }
146 }
```

*(End of definition for `\_mark_update_structure_from_splitmarks:n`.)*

```
__mark_get_marks_for_reinsertion:nNN
```

This function extracts the marks from the material in the first argument but it does not update any the mark structures. Instead, it collects the marks in the token lists given as the second and third argument, in such a way that they can be reinserted by just executing the token lists.<sup>47</sup>

```
147 \cs_new_protected:Npn __mark_get_marks_for_reinsertion:nNN #1#2#3 {
```

First we clear the temporary token lists as we haven't seen any marks yet.

```
148 \tl_clear:N \g__mark_first_marks_tl
149 \tl_clear:N \g__mark_last_marks_tl
```

Then we extract all top-level marks, thereby filling the token lists with suitable `\mark_insert:nn` calls.

```
150 __mark_extract_and_handle_marks:nn
```

The first argument holds the code for fill the token lists and the second is the material we extract from.

```
151 __mark_get_from_splitmarks:
152 { #1 }
```

Finally, we copy the updated (or not updated) temporary token lists to the two that have been supplied when the function was called. By convention “get” operations return their values in local variables and `\_\_mark_extract_and_handle_marks:nn` runs in a group, which is why we have to use global temporary variables for collecting.

```
153 \tl_set_eq:NN #2 \g__mark_first_marks_tl
154 \tl_set_eq:NN #3 \g__mark_last_marks_tl
155 }
```

(End of definition for `\_\_mark_get_marks_for_reinsertion:nNN`.)

```
__mark_get_from_splitmarks:
```

This function is called after we have done a `\vsplit` to update the split marks. It loops through all mark classes to find out if there are marks for this class and if so updates the global tls used for collecting.

```
156 \cs_new_protected:Npn __mark_get_from_splitmarks: {
157 \seq_map_inline:Nn \g__mark_classes_seq
158 {
```

First we to get the last mark for the current class from the material supplied.

```
159 \tl_gset:No \g__mark_tmp_tl
160 { \tex_splitbotmarks:D \use:c { c__mark_class_\#\#1_mark } }
```

If this mark doesn't exist then obviously first mark doesn't either, so we do nothing (other than issuing some debugging info).

We have to be a little careful here: something like `\mark_insert:nn{foo}{}{}` adds an “empty” mark that we should not confuse with the case where there is no mark at all.

When there is no mark at all we get a truly empty `\g\_\_mark_tmp_tl` as a result. This is why we have to make sure that an “empty” mark generated with `\mark_insert:nn` only appears to be empty when it is typeset, but fails the next test (see below how this is done).

```
161 \tl_if_empty:NTF \g__mark_tmp_tl
162 {
163 (*trace)
```

---

<sup>47</sup>It is probably enough to collect everything in a single token list as long as we put the first marks first and the last marks last). But for extra flexibility, I currently use 2 token lists. This might change when it is really clear that this is never needed.

```

164 __mark_debug:n { \iow_term:x { Marks:~no~ marks~
165 for~ class~ '##1'~\msg_line_context: } }
166
```

If it wasn't empty, i.e., if it had a real value then we use this value for our new last mark instead. This means we put an appropriate `\mark_insert:nn` statement into `\g__mark_last_marks_tl`.

```

168 {
169
```

```
170 __mark_debug:n { \iow_term:x { Marks:~ extract~ last~
171 mark~ for~ class~ '##1'~ =~ \g__mark_tmp_tl } }
172
```

```
173 \tl_gput_right:Ne \g__mark_last_marks_tl
174 { \mark_insert:nn {##1} { \g__mark_tmp_tl } }
```

Because we had a last mark we also have a first mark (which might be the same, but might not be), so we pick that up and add it to the `\g__mark_first_marks_tl` token list. This explains why we first checked for the last mark because that makes the processing faster in case there is none.

```

175
```

```
176 __mark_debug:n { \iow_term:x { Marks:~ extract~ first~
177 mark~ for~ class~ '##1'~ =~
178 \tex_splitfirstmarks:D
179 \use:c { c__mark_class_##1_mark }
180 } }
181
```

```
182 \tl_gput_right:Ne \g__mark_first_marks_tl
183 { \mark_insert:nn {##1}
184 {
185 \tex_splitfirstmarks:D
186 \use:c { c__mark_class_##1_mark }
187 }
188 }
189 }
190 }
191 }
```

*(End of definition for `\__mark_get_from_splitmarks:..`)*

`\g__mark_first_marks_tl` These are two global temporary variables used in the code above.

`\g__mark_last_marks_tl`

```
192 \tl_new:N \g__mark_first_marks_tl
193 \tl_new:N \g__mark_last_marks_tl
```

*(End of definition for `\g__mark_first_marks_tl` and `\g__mark_last_marks_tl`.)*

`\__mark_update_structure_alias:nn` This function copies the structure for one region to another (name), e.g., from `page` to `previous-page` above, or later from `column` to `first-column`, etc.

```
194 \cs_new_protected:Npn __mark_update_structure_alias:nn #1#2 {
```

This requires a simple loop through all mark classes copying the token list from one name to the next.

```

195 \seq_map_inline:Nn \g__mark_classes_seq
196 {
197 \tl_gset_eq:cc { g__mark_ #1 _top_ ##1 _tl }
```

```

198 { g__mark_ #2 _top_ ##1 _tl }
199 \tl_gset_eq:cc { g__mark_ #1 _first_ ##1 _tl }
200 { g__mark_ #2 _first_ ##1 _tl }
201 \tl_gset_eq:cc { g__mark_ #1 _last_ ##1 _tl }
202 { g__mark_ #2 _last_ ##1 _tl }
203 }
204 }
```

(End of definition for `\_mark_update_structure_alias:nn`.)

`\_mark_update_structure_to_err:n`

`\_mark_error:n`

```

205 \cs_new_protected:Npn _mark_update_structure_to_err:n #1 {
206 \seq_map_inline:Nn \g__mark_classes_seq
207 {
208 \tl_gset:cn { g__mark_ #1 _top_ ##1 _tl } { _mark_error:n {#1} }
209 \tl_gset:cn { g__mark_ #1 _first_ ##1 _tl } { _mark_error:n {#1} }
210 \tl_gset:cn { g__mark_ #1 _last_ ##1 _tl } { _mark_error:n {#1} }
211 }
212 }
```

Given that this is used in only one place, we could hardwire the argument which would be a bit more compact, but who knows, perhaps we end up with another reason to use this error command elsewhere, so for now we keep the argument.

```

213 \cs_new_protected:Npn _mark_error:n #1 {
214 \msg_error:nnn { mark } { invalid-use } {#1}
215 }
```

(End of definition for `\_mark_update_structure_to_err:n` and `\_mark_error:n`.)

### 7.3 Placing and retrieving marks

`\mark_insert:nn`

This function puts a mark for some  $\langle class \rangle$  at the current point.

```

216 \cs_new_protected:Npn \mark_insert:nn #1#2
217 {
218 \seq_if_in:NnTF \g__mark_classes_seq {#1}
219 {
```

We need to pass the evaluated argument into the mark but protected commands should not expand including those protected using the `\protect` approach of L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub>. We also disable `\label` and the like.<sup>48</sup>

At this point the code eventually should get a public (and a kernel) hook instead of a set of hardwired settings.

```

220 \group_begin:
```

Within the group we alter some comments, e.g., `\label` or `\index`, to do the right at this point. This is done in the kernel hook `\@kernel@before@insertmark` which is followed by the public hook `insertmark` that can be used by packages to augment or alter that setup as necessary.

```

221 \@kernel@before@insertmark
222 \hook_use:n { insertmark }
223 \unrestored@protected@xdef \g__mark_tmp_tl
224 {
```

---

<sup>48</sup>Straight copy from `latex.ltx` but is this even correct? At least a label in a running header makes little sense if it gets set several times! Maybe that needs looking at in the 2e kernel.

To ensure that marks are unique we insert a hidden sequence marker at the beginning of the content of the mark containing the sequence number of the mark.

```

225 __mark_id:n{ \int_use:N\g__mark_int }
226 #2
227 }
228 <*trace>
229 __mark_debug:n{ \iow_term:x { Marks:~ set~#1~-<-
230 '\tl_to_str:V \g__mark_tmp_tl' ~ \msg_line_context: } }
231 </trace>
232 \tex_marks:D \use:c { c__mark_class_ #1 _mark }
233 {

```

Here is the trick to avoid truly empty marks: if the result from the above processing is empty we add something which eventually becomes empty, but not immediately; otherwise we just put `\g__mark_tmp_tl` in.

```

234 % this is no longer needed with 1.0f
235 %
236 % \tl_if_empty:NTF \g__mark_tmp_tl
237 % { \exp_not:n { \prg_do_nothing: } }
238 % { \exp_not:o { \g__mark_tmp_tl } }
239 % \exp_not:o { \g__mark_tmp_tl }
240 }
241 \group_end:

```

A mark introduces a possible break point and in certain situations that should not happen in vertical mode in L<sup>A</sup>T<sub>E</sub>X. This needs some cleanup ....

```

241 \if@nobreak\ifvmode\nobreak\fi\fi
242 }

```

If the mark class was not known, raise an error.

```

243 {
244 \msg_error:nnx { mark } { unknown-class }
245 { \tl_to_str:n {#1} }
246 }
247 }

```

(End of definition for `\mark_insert:nn`. This function is documented on page 943.)

`\__mark_id:n` A hidden marker is placed into every mark added by `\mark_insert:nn`. It will not show up in the output but its argument (a counter value that is incremented) makes all marks unique so the test for “equal” is not fooled by two different marks having the same mark text.

```

248 \cs_new_protected:Npn __mark_id:n #1 { }

```

(End of definition for `\__mark_id:n`.)

By default `\label`, `\index`, and `\glossary` do nothing when the mark is inserted.

```

249 \int_new:N \g__mark_int
250 \cs_new:Npn \@kernel@before@insertmark {
251 \cs_set_eq:NN \label \scan_stop:
252 \cs_set_eq:NN \index \scan_stop:
253 \cs_set_eq:NN \glossary \scan_stop:

```

We count each mark and use that to place a hidden marker in front of the mark text. To ensure that there is no overflow (very unlikely but you never know) we restart every 100000 marks. Thus, if somebody puts more than that number of marks on a single page you could construct a scenario in which that approach fails.

```

254 \int_compare:nNnTF \g__mark_int < {99999}
255 { \int_gincr:N \g__mark_int }
256 { \int_gzero:N \g__mark_int }
257
258 }
```

The public hook to augment the setup.

```
259 \hook_new:n {insertmark}
```

*(End of definition for \@kernel@before@insertmark and insertmark.)*

\mark\_use\_top:nn  
\mark\_use\_first:nn  
\mark\_use\_last:nn

To retrieve the first, last or top region mark, we grab the appropriate value stored in the corresponding token list variable and pass its contents back. These functions should be used only in output routines and only after \\_\_mark\_update\_structure\_from\_material:nn has acted, otherwise their value will be wrong.

If used with an unknown class or region they generate an error (fairly low-level because we are in an expandable context).

Each mark starts with an id and while the id does not print it is nevertheless better to remove it when returning the mark, so that downstream manipulation of the data doesn't have to deal with it.

```

260 \cs_new:Npn \mark_use_first:nn #1#2 { __mark_use:v { g__mark_#1_first #2_tl } }
261 \cs_new:Npn \mark_use_last:nn #1#2 { __mark_use:v { g__mark_#1_last #2_tl } }
262 \cs_new:Npn \mark_use_top:nn #1#2 { __mark_use:v { g__mark_#1_top #2_tl } }
```

This is what the \use\_none:nn accomplishes.

```

263 \cs_new:Npn __mark_use:n #1 { \exp_not:o { \use_none:nn #1 } }
264 \cs_generate_variant:Nn __mark_use:n { v }
```

*(End of definition for \mark\_use\_top:nn, \mark\_use\_first:nn, and \mark\_use\_last:nn. These functions are documented on page 944.)*

## 7.4 Comparing mark values

\mark\_if\_eq:nnnnTF  
\mark\_if\_eq:nnnnnnTF

Test if in a given region (#1) for a given class (#2) the marks in position #3 and #4 (top, first, or last) are identical

```

265 \prg_new_conditional:Npnn \mark_if_eq:nnnn #1#2#3#4 { T , F , TF }
266 {
267 \tl_if_eq:ccTF { g__mark_ #1 _#3_ #2 _tl }
268 { g__mark_ #1 _#4_ #2 _tl }
269 \prg_return_true:
270 \prg_return_false:
271 }
```

The fully general test (with two triplets of the form *<region>*, *<class>*, and *<position>*) is this:

```

272 \prg_new_conditional:Npnn \mark_if_eq:nnnnnn #1#2#3#4#5#6 { T , F , TF }
273 {
274 \tl_if_eq:ccTF { g__mark_ #1 _#3_ #2 _tl }
275 { g__mark_ #4 _#6_ #5 _tl }
276 \prg_return_true:
```

```

277 \prg_return_false:
278 }
(End of definition for \mark_if_eq:nnnnTF and \mark_if_eq:nnnnnnTF. These functions are
documented on page 945.)
```

## 7.5 Messages

Mark errors are L<sup>A</sup>T<sub>E</sub>X kernel errors:

```

279 \prop_gput:Nnn \g_msg_module_type_prop { mark } { LaTeX }
280 \msg_new:nnn { mark } { class-already-defined }
281 { Mark-class-'#1'-already-defined }
282 {
283 \c__msg_error_text_tl
284 LaTeX-was-asked-to-define-a-new-mark-class-called-'#1':-
285 this-mark-class-already-exists.
286 \c__msg_return_text_tl
287 }
288 \msg_new:nnn { mark } { unknown-class }
289 { Unknown-mark-class-'#1'. }
290 {
291 \c__msg_error_text_tl
292 LaTeX-was-asked-to-manipulate-a-mark-of-class-'#1',-
293 but-this-class-of-marks-does-not-exist.
294 }
295
296 \msg_new:nnn { mark } { invalid-use }
297 { Mark-region-'#1'-not ~usable }
298 {
299 \c__msg_error_text_tl
300 The-region-'#1'-can-only-be-used-after-
301 all-columns-have-been-assembled.
302 \c__msg_return_text_tl
303 }
```

## 7.6 Debugging the mark structures

Code and commands in this section are not final, it needs more experimentation to see what kind of tracing information is going to be useful in practice. For now the tracing is mainly meant to be used for code testing and not so much for application testing.

It is quite likely that the commands and the behavior of the tracing might change in the future once we gained some experience with it.

|                     |                                    |
|---------------------|------------------------------------|
| \g__mark_debug_bool | Holds the current debugging state. |
|---------------------|------------------------------------|

```

304 \bool_new:N \g__mark_debug_bool
(End of definition for \g__mark_debug_bool.)
```

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| \mark_debug_on: | Turns debugging on and off by redefining \__mark_debug:n. |
|-----------------|-----------------------------------------------------------|

|                  |                                               |
|------------------|-----------------------------------------------|
| \mark_debug_off: | 305 \cs_new_eq:NN \__mark_debug:n \use_none:n |
|------------------|-----------------------------------------------|

|                 |                                           |
|-----------------|-------------------------------------------|
| \__mark_debug:n | 306 \cs_new_protected:Npn \mark_debug_on: |
|-----------------|-------------------------------------------|

|                     |       |
|---------------------|-------|
| \__mark_debug_gset: | 307 { |
|---------------------|-------|

```

308 \bool_gset_true:N \g__mark_debug_bool
309 __mark_debug_gset:
310 }
311 \cs_new_protected:Npn \mark_debug_off:
312 {
313 \bool_gset_false:N \g__mark_debug_bool
314 __mark_debug_gset:
315 }
316 \cs_new_protected:Npn __mark_debug_gset:
317 {
318 \cs_gset_protected:Npx __mark_debug:n ##1
319 { \bool_if:NT \g__mark_debug_bool {##1} }
320 }

```

(End of definition for `\mark_debug_on`: and others. These functions are documented on page 946.)

`\DebugMarksOn` CamelCase commands for debugging.

`\DebugMarksOff`

```

321 \cs_new_eq:NN \DebugMarksOn \mark_debug_on:
322 \cs_new_eq:NN \DebugMarksOff \mark_debug_off:

```

(End of definition for `\DebugMarksOn` and `\DebugMarksOff`. These functions are documented on page 946.)

`\__mark_class_status:nn` Shows the mark values across all regions for one mark class (#2). The first argument gives some `<info>` to help identifying where the command was called.

```

323 (*trace)
324 \cs_new_protected:Npn __mark_class_status:nn #1#2
325 {
326 \typeout{ Marks:#2~ #1:}
327 \typeout{@spaces page~ (current):
328 | \exp_not:v { g__mark_page_top_ #2 _tl }
329 | \exp_not:v { g__mark_page_first_ #2 _tl }
330 | \exp_not:v { g__mark_page_last_ #2 _tl } |}
331 \typeout{@spaces page~ (previous):
332 | \exp_not:v { g__mark_previous-page_top_ #2 _tl }
333 | \exp_not:v { g__mark_previous-page_first_ #2 _tl }
334 | \exp_not:v { g__mark_previous-page_last_ #2 _tl } |}
335 \typeout{@spaces column~ (previous):
336 | \exp_not:v { g__mark_previous-column_top_ #2 _tl }
337 | \exp_not:v { g__mark_previous-column_first_ #2 _tl }
338 | \exp_not:v { g__mark_previous-column_last_ #2 _tl } |}
339 \typeout{@spaces column~ (current):
340 | \exp_not:v { g__mark_column_top_ #2 _tl }
341 | \exp_not:v { g__mark_column_first_ #2 _tl }
342 | \exp_not:v { g__mark_column_last_ #2 _tl } |}
343 \typeout{@spaces column~ (first):
344 | \exp_not:v { g__mark_first-column_top_ #2 _tl }
345 | \exp_not:v { g__mark_first-column_first_ #2 _tl }
346 | \exp_not:v { g__mark_first-column_last_ #2 _tl } |}
347 \typeout{@spaces column~ (second):
348 | \exp_not:v { g__mark_last-column_top_ #2 _tl }
349 | \exp_not:v { g__mark_last-column_first_ #2 _tl }
350 | \exp_not:v { g__mark_last-column_last_ #2 _tl } |}
351 }

```

(End of definition for `\__mark_class_status:nn`.)

`\__mark_status:n` Show a snapshot of all mark class values across all regions.

```
352 \cs_new_protected:Npn __mark_status:n #1
353 {
354 \seq_map_inline:Nn \g__mark_classes_seq
355 { __mark_class_status:nn {#1} {##1} }
356 }
357
```

(End of definition for `\__mark_status:n`.)

`\ShowMarksAt` Debugging helper that displays a snapshot of all known mark structures. The argument is a text string that is displayed to help identifying when the snapshot was made.

This may not stay like this (or at all), which is why it isn't yet documented as an official command.

```
358 \cs_new_protected:Npn \ShowMarksAt #1 {
359 (*trace)
360 __mark_debug:n { __mark_status:n {#1} }
361
```

(End of definition for `\ShowMarksAt`.)

## 7.7 Designer-level interfaces

`\NewMarkClass` These two are identical to the L3 programming layer commands.

`\InsertMark`

```
363 \cs_new_eq:NN \NewMarkClass \mark_new_class:n
364 @onlypreamble \NewMarkClass
365 \cs_new_eq:NN \InsertMark \mark_insert:nn
```

(End of definition for `\NewMarkClass` and `\InsertMark`. These functions are documented on page 943.)

`\TopMark` The following commands take an optional argument that defaults to page. There is no checking that the region is actually valid. If not there is simply an empty return.

`\FirstMark`

`\LastMark`

```
366 \NewExpandableDocumentCommand \FirstMark { O{page} m }
367 { \mark_use_first:nn {#1}{#2} }
368 \NewExpandableDocumentCommand \LastMark { O{page} m }
369 { \mark_use_last:nn {#1}{#2} }
370 \NewExpandableDocumentCommand \TopMark { O{page} m }
371 { \mark_use_top:nn {#1}{#2} }
```

(End of definition for `\TopMark`, `\FirstMark`, and `\LastMark`. These functions are documented on page 944.)

`\IfMarksEqualTF` We only provide CamelCase commands for the case with one region (optional) and one class. One could think of also providing a version for the general case with several optional arguments, but use cases for this are most likely rare, so not done yet.

`\IfMarksEqualT`

```
372 \NewExpandableDocumentCommand \IfMarksEqualTF {O{page}mmm} {
373 \mark_if_eq:nnnnTF {#1}{#2}{#3}{#4}
374 }
375 \NewExpandableDocumentCommand \IfMarksEqualT {O{page}mmm} {
376 \mark_if_eq:nnnnT {#1}{#2}{#3}{#4}
```

```

377 }
378 \NewExpandableDocumentCommand \IfMarksEqualF {0{page}mmm} {
379 \mark_if_eq:nnnnF {#1}{#2}{#3}{#4}
380 }

```

(End of definition for `\IfMarksEqualTF`, `\IfMarksEqualT`, and `\IfMarksEqualF`. These functions are documented on page 945.)

## 8 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> integration

### 8.1 Core L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> integration

`\_mark_update_singlecol_structures:`

This command updates the mark structures if we are producing a single column document.

```

381 \cs_new_protected:Npn _mark_update_singlecol_structures: {

```

First we update the `page` region (which also updates the `previous-page`.

The `\@outputbox` is normally in `\vbox` in L<sup>A</sup>T<sub>E</sub>X but we can't take that for granted (an `amsmath` test document changed it to an `\hbox` just to trip me up) so we are a little careful with unpack now.

```

382 \box_if_vertical:NTF \@outputbox
383 {
384 _mark_update_structure_from_material:nn {page}
385 { \vbox_unpack:N \@outputbox }
386 }
387 {
388 _mark_update_structure_from_material:nn {page}
389 { \hbox_unpack:N \@outputbox }
390 }

```

The we provide the necessary updates for the aliases.

```

391 _mark_update_structure_alias:nn {previous-column}{previous-page}
392 _mark_update_structure_alias:nn {column}{page}
393 _mark_update_structure_alias:nn {first-column}{page}
394 _mark_update_structure_alias:nn {last-column}{page}
395 (*trace)
396 % move this into status itself?
397 _mark_debug:n
398 {
399 _mark_status:n
400 { in~ OR~ (
401 \legacy_if:nTF {@twoside}
402 { twoside-
403 \int_if_odd:nTF \c@page
404 { odd }{ even }
405 }
406 { oneside }
407)
408 }
409 }
410 (/trace)
411 }

```

(End of definition for `\_mark_update_singlecol_structures:..`)

\\_\\_mark\\_update\\_dblcol\\_structures: This commands handles the updates if we are doing two-column pages.  
 412 \cs\_new\_protected:Npn \\_\\_mark\_update\_dblcol\_structures: {  
 First we update the `column` and `previous-column` regions using the material assembled in `\@outputbox`.

```

 413 \box_if_vertical:NTF \@outputbox
 414 {
 415 __mark_update_structure_from_material:nn {column}
 416 { \vbox_unpack:N \@outputbox }
 417 }
 418 {
 419 __mark_update_structure_from_material:nn {column}
 420 { \hbox_unpack:N \@outputbox }
 421 }

```

How we have to update the alias regions depends on whether or not `\@opcol` was called to process the first column or to produce the completed page

```

 422 \legacy_if:nTF {@firstcolumn}
 423 {

```

If we are processing the first column then `column` is our `first-column` and there is no `last-column` yet, so we make those an error.

```

 424 __mark_update_structure_alias:nn {first-column}{column}
 425 __mark_update_structure_to_err:n {last-column}
 426 }
 427 {

```

If we produce the completed page then the `first-column` is the same as the new `previous-column`. However, the structure should already be correct if you think about it (because it was set to `column` last time which is now the `previous-column`), thus there is no need to make an update.

```

 428 % __mark_update_structure_alias:nn {first-column}{previous-column}

```

However, we now have a proper `last-column` so we assign that.

```

 429 __mark_update_structure_alias:nn {last-column}{column}

```

What now remains doing is to update the `page` and `previous-page` regions. For this we have to copy the settings in `page` into `previous-page` and then update `page` such that the top and first marks are taken from the `first-column` region and the last marks are taken from the `last-column` region. All this has to be done for all mark classes so we loop over our sequence.

Note that one loop is needed if we arrange the copy statements in a suitable way.

```

 430 \seq_map_inline:Nn \g_mark_classes_seq
 431 {

```

The `previous-page` updates need to come before the updates for `page` region because otherwise the values to copy are already overwritten. necessary values.

```

 432 \tl_gset_eq:cc { g_mark_previous-page_top_ ##1 _tl }
 433 { g_mark_page_top_ ##1 _tl }
 434 \tl_gset_eq:cc { g_mark_previous-page_first_ ##1 _tl }
 435 { g_mark_page_first_ ##1 _tl }
 436 \tl_gset_eq:cc { g_mark_previous-page_last_ ##1 _tl }
 437 { g_mark_page_last_ ##1 _tl }

```

To update the `top` we only have to copy what is in `first-column`:

```
438 \tl_gset_eq:cc { g__mark_page_top_ ##1 _tl }
439 { g__mark_first-column_top_ ##1 _tl }
440
```

Updating the `first` mark for the `page` region is more complicated. We first have to find out of there is any mark in the first column (this can be done by comparing the `top` and the `first` mark of that region).

```
441 \tl_if_eq:ccTF { g__mark_first-column_top_ ##1 _tl }
442 { g__mark_first-column_first_ ##1 _tl }
443 }
```

If there is no mark in the first column we copy the first mark of the last column. If that doesn't contain a mark we still get the right result because the first mark is then equal to the top mark.

```
444 \tl_gset_eq:cc { g__mark_page_first_ ##1 _tl }
445 { g__mark_last-column_first_ ##1 _tl }
446 }
447 {
```

On the other hand, if there is a mark in the first column we copy over the `first` mark from that column.

```
448 \tl_gset_eq:cc { g__mark_page_first_ ##1 _tl }
449 { g__mark_first-column_first_ ##1 _tl }
450 }
```

The logic for the `last` page mark is again simple, we can just copy the value in the `last` mark of the last column. If that column doesn't contain any marks, then the value in `last` will be automatically the same as the `last` from the first column.

```
451 \tl_gset_eq:cc { g__mark_page_last_ ##1 _tl }
452 { g__mark_last-column_last_ ##1 _tl }
453 }
454 }
455 <*trace>
456 __mark_debug:n
457 {
458 __mark_status:n
459 { in~ OR~ (
460 \legacy_if:nTF {@twoside}
461 { twoside-
462 \int_if_odd:nTF \c@page
463 { odd }{ even }
464 }
465 { oneside }
466 \space
467 \legacy_if:nTF {@firstcolumn}
468 { first~ }{ second~ }
469 column)
470 }
471 }
472 </trace>
473 }
```

(End of definition for `\__mark_update dblcol_structures:..`)

```
474 <@=>
```

```

\@expl@@@mark@update@singlecol@structures@@
475 \cs_new_eq:NN \@expl@@@mark@update@singlecol@structures@@
476 __mark_update_singlecol_structures:

(End of definition for \@expl@@@mark@update@singlecol@structures@@.)

\@expl@@@mark@update@dblcol@structures@@
477 \cs_new_eq:NN \@expl@@@mark@update@dblcol@structures@@
478 __mark_update_dblcol_structures:

(End of definition for \@expl@@@mark@update@dblcol@structures@@.)

```

## 8.2 Other L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\epsilon$</sub> output routines

This section will cover `multicol` and other packages altering or providing their own output routine. Not done yet.

## 8.3 Rollback information

```

479 \begin{macro}{\IncludeInRelease[0000/00/00]{ltmarks}}
480 \begin{macro}{\UndoMarksHandling}
481 \begin{macro}{\NewMarkClass[1]}

```

We keep the interface commands around even if we roll back in case they are used in packages that don't roll back. Not likely to do a lot of good, but then there is not much we can do, but this at least they won't give unknown csname errors.

```

482 \begin{macro}{\DeclareRobustCommand \NewMarkClass[1]}
483 \begin{macro}{\DeclareRobustCommand \InsertMark[2]}
484 \begin{macro}{\RenewExpandableDocumentCommand \FirstMark { O{} m } { }}
485 \begin{macro}{\RenewExpandableDocumentCommand \LastMark { O{} m } { }}
486 \begin{macro}{\RenewExpandableDocumentCommand \TopMark { O{} m } { }}
487 \begin{macro}{\RenewExpandableDocumentCommand \IfMarksEqualTF { O{} mmm }{}}
488 \end{macro}

```

Same here, this avoided extra roll back code in the OR.

```

489 \begin{macro}{\let \Expl@@@mark@update@singlecol@structures@@ \relax}
490 \begin{macro}{\let \Expl@@@mark@update@dblcol@structures@@ \relax}
491 \begin{macro}{\EndModuleRelease}
492 \begin{macro}{\ExplSyntaxOff}
493 \end{macro}

```

Reset module prefix:

```

496 \COC=

```

# File 49

## 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 \cmkboth command, which is \let by the pagestyle command (\ps@...) to \markboth for setting the heading or to \gobbletwo to do nothing.

1 <\*2ekernel>

\pagestyle User command to set the page style for this and following pages.

```
2 \def\pagestyle#1{%
3 \ifundefined{ps@#1}%
4 \undefinedpagestyle
5 {\@nameuse{ps@#1}}}
```

(End of definition for \pagestyle.)

\thispagestyle User command to set the page style for this page only.

```
6 \def\thispagestyle#1{%
7 \ifundefined{ps@#1}%
8 \undefinedpagestyle
9 {\global\@specialpagetrue\gdef\@specialstyle{#1}}}
```

(End of definition for \thispagestyle.)

\ps@empty The empty page style: No head or foot line.

```
10 \def\ps@empty{%
11 \let\@mkboth\@gobbletwo\let\@oddhead\@empty\let\@oddfoot\@empty
12 \let\@evenhead\@empty\let\@evenfoot\@empty}
```

(End of definition for \ps@empty.)

\ps@plain The plain page style: No head, centred page number in foot.

```
13 \def\ps@plain{\let\@mkboth\@gobbletwo
14 \let\@oddhead\@empty\def\@oddfoot{\reset@font\hfil\thepage
15 \hfil}\let\@evenhead\@empty\let\@evenfoot\@oddfoot}
```

(End of definition for \ps@plain.)

\@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.

```
16 \let\@leftmark\@firstoftwo
17 \let\@rightmark\@secondoftwo
```

(End of definition for \@leftmark and \@rightmark.)

```
18 {/2ekernel}
19 {*2ekernel | latexrelease}
20 {latexrelease}\IncludeInRelease{2022/06/01}%
21 {latexrelease} {\markboth}{New mark support}%
```

\markboth User commands for setting L<sup>A</sup>T<sub>E</sub>X marks.

\markright Test for \nobreak added 15 Apr 86 in \markboth and \markright letting \label and \index to \relax added 22 Feb 86 so these commands can appear in sectioning command arguments RmS 91/06/21 Same for \glossary

```
22 \ExplSyntaxOn
23 \DeclareRobustCommand*\markboth[2]{%
24 \begingroup
25 \let\label\relax \let\index\relax \let\glossary\relax
26 \unrestored@protected\xdef\@themark {{\#1}{\#2}}%
27 \temptokena\expandafter{\@themark}}
```

In addition to generating the legacy mark we output the individual ones as well at the very same point. The legacy mark is kept unchanged in order to work with packages that expect that mark in exactly the way it is right now.

We might want to think about how to improve this in one-side documents, see comments below.

We have not changed all of the code to L3 prog layer convention, in case packages attempt to do some patching and expect the 2e names being around. Eventually this should and will change.

```
28 \mark_insert:nn{2e-left}{\#1}
29 \mark_insert:nn{2e-right}{\#2}
```

```

30 \tl_if_empty:nF{#2}{ \mark_insert:nn{2e-right-nonempty}{#2} }
31 \mark{\the\@temptokena}%
32 \endgroup
33 \if@nobreak\ifvmode\nobreak\fi\fi}

34 \DeclareRobustCommand*\markright[1]{%
35 \begingroup
36 \let\label\relax \let\index\relax \let\glossary\relax

```

Protection is handled inside \markright.

```

37 \expandafter\markright\@themark {#1}%
38 \@temptokena \expandafter{\@themark}%

```

Same game with \markright more or less ...

```

39 \mark_insert:nn{2e-right}{#1}
40 \tl_if_empty:nF{#1}{ \mark_insert:nn{2e-right-nonempty}{#1} }

```

The legacy L<sup>A</sup>T<sub>E</sub>X mechanism always sets left and right mark, i.e., if a sub-mark (i.e., right mark) is set the corresponding main mark also is getting a mark with the same value it had previously. However, for the individual mark classes this means we are losing information so for them that is not done.

```

41 % \mark_insert:nn{2e-left}{\exp_after:wN \use_i:nn \@themark }
42 \mark{\the\@temptokena}%
43 \endgroup
44 \if@nobreak\ifvmode\nobreak\fi\fi}
45 \ExplSyntaxOff

```

(End of definition for \markboth and \markright. These functions are documented on page 946.)

```

46 </2ekernel | latexrelease>
47 <latexrelease>\EndIncludeInRelease
48 <latexrelease>\IncludeInRelease{2019/10/01}%
49 <latexrelease> {\markboth}{Make commands robust}%
50 <latexrelease>
51 <latexrelease>\DeclareRobustCommand*\markboth[2]{%
52 <latexrelease> \begingroup
53 <latexrelease> \let\label\relax \let\index\relax \let\glossary\relax
54 <latexrelease> \unrestored@protected@xdef@\themark {\#1}{\#2}%
55 <latexrelease> \@temptokena \expandafter{\@themark}%
56 <latexrelease> \mark{\the\@temptokena}%
57 <latexrelease> \endgroup
58 <latexrelease> \if@nobreak\ifvmode\nobreak\fi\fi}
59 <latexrelease>\DeclareRobustCommand*\markright[1]{%
60 <latexrelease> \begingroup
61 <latexrelease> \let\label\relax \let\index\relax \let\glossary\relax
62 <latexrelease> \expandafter\markright\@themark {#1}%
63 <latexrelease> \@temptokena \expandafter{\@themark}%
64 <latexrelease> \mark{\the\@temptokena}%
65 <latexrelease> \endgroup
66 <latexrelease> \if@nobreak\ifvmode\nobreak\fi\fi}
67 <latexrelease>
68 <latexrelease>\EndIncludeInRelease
69 <latexrelease>\IncludeInRelease{0000/00/00}%
70 <latexrelease> {\markboth}{Make commands robust}%
71 <latexrelease>
72 <latexrelease>\kernel@make@fragile\markboth

```

```

73 ⟨latexrelease⟩\kernel@make@fragile\markright
74 ⟨latexrelease⟩
75 ⟨latexrelease⟩\EndIncludeInRelease
76 ⟨*2ekernel⟩

\@markright
\leftmark
\rightmark
77 \def\@markright#1#2#3{\@temptokena {#1}%
78 \unrestored@protected@xdef\@themark{{\the\@temptokena}{#3}}}
79 \def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
80 \def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}

(End of definition for \@markright, \leftmark, and \rightmark.)

\@themark Initialise LATEX's marks without setting a TEX mark ⟨whatsit⟩.
81 \def\@themark{}{ }

(End of definition for \@themark.)

\mark Test versions of LATEX 2E initialised TEX's \mark system at this point, but this was
removed before the first release.

AtBeginDocument{\mark{}{}}

(End of definition for \mark.)

\raggedbottom \raggedbottom typesets pages with no vertical stretch, so they have their natural height
instead of all being exactly the same height. (Uses a space of .0001fil to avoid interfering
with the 1fil space of \newpage.)
82 \DeclareRobustCommand\raggedbottom{%
83 \def\@textbottom{\vskip \z@ \oplus.0001fil}\let\@texttop\relax}

(End of definition for \raggedbottom.)

\flushbottom \flushbottom: Inverse of \raggedbottom — makes all pages the same height.
84 \DeclareRobustCommand\flushbottom{%
85 \let\@textbottom\relax \let\@texttop\relax}

(End of definition for \flushbottom.)

\sloppy \sloppy will never (well, hardly ever) produce overfull boxes, but may produce underfull
ones. (14 June 85)
86 \DeclareRobustCommand\sloppy{%
87 \tolerance 9999%
88 \emergencystretch 3em%
89 \hfuzz .5\p@
90 \vfuzz\hfuzz}

(End of definition for \sloppy.)

\slloppypar (env.) A slloppypar environment is equivalent to {\par \sloppy ... \par}.
91 \def\slloppypar{\par\sloppy}
92 \def\endslloppypar{\par}

```

**\fussy** Resets T<sub>E</sub>X's parameters to their normal finicky values.

```
93 \DeclareRobustCommand\fussy{%
94 \emergencystretch\z@
95 \tolerance 200%
96 \hfuzz .1\p@
97 \vfuzz\hfuzz}
```

(End of definition for **\fussy**.)

**\overfullrule** L<sup>A</sup>T<sub>E</sub>X default is no overfull box rule. Changed by document class option.

```
98 \overfullrule 0pt
```

(End of definition for **\overfullrule**.)

```
99 </2ekernel>
```

# File 50

## ltclass.dtx

### 1 Introduction

This file implements the following declarations, which replace `\documentstyle` in L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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 L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  and L<sup>A</sup>T<sub>E</sub>X2.09 is that L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  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 L<sup>A</sup>T<sub>E</sub>X2.09 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 L<sup>A</sup>T<sub>E</sub>X2.09 compatibility mode. As far as most packages are concerned, this only affects the warnings and errors L<sup>A</sup>T<sub>E</sub>X 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 (env.)`

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.

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.

`\IfPackageLoadedTF` To find out if a package has already been loaded, use  
`\IfClassLoadedTF` `\IfPackageLoadedTF{(package)}{(true)}{(false)}`  
`\@ifpackageloaded` or the old name `\@ifpackageloaded`.  
`\IfPackageAtLeastTF` To find out if a package has already been loaded with a version equal to or more recent than `(date)`, use  
`\IfClassAtLeastTF` `\IfPackageAtLeastTF{(package)}{(date)}{(true)}{(false)}`  
`\IfFileAtLeastTF` `\@ifpackagelater` or the old name `\@ifpackagelater`.  
`\@ifclasslater` or the old name `\@ifclasslater`.  
`\IfFormatAtLeastTF` To test the format date use  
`\IfFormatAtLeastTF{(date)}{(true)}{(false)}`

`\IfPackageLoadedWithOptionsTF` To find out if a package has already been loaded with at least the options `(options)`,  
`\IfClassLoadedWithOptionsTF` use  
`\@ifpackagewith` `\IfPackageLoadedWithOptionsTF{(package)}{(options)}{(true)}{(false)}`  
`\@ifclasswith` or the old name `\@ifpackagewith`.

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

|                                 |                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\InputIfFileExists</code> | <code>\InputIfFileExists{\file}{\then}{\else}</code>                                                                                                                                                                                                                                                                                                          |
|                                 | Inputs <code>\file</code> if it exists. Immediately before the input, <code>\then</code> is executed. Otherwise <code>\else</code> is executed.                                                                                                                                                                                                               |
| <code>\IfExists</code>          | As above, but does not input the file.                                                                                                                                                                                                                                                                                                                        |
|                                 | One thing you might like to put in the <code>\else</code> clause is                                                                                                                                                                                                                                                                                           |
| <code>\@missingfileerror</code> | This starts an interactive request for a filename, supplying default extensions. Just hitting return causes the whole input to be skipped and entering <code>x</code> quits the current run,                                                                                                                                                                  |
| <code>\input</code>             | This has been redefined from the L <sup>A</sup> T <sub>E</sub> X2.09 definition, in terms of the new commands <code>\InputIfFileExists</code> and <code>\@missingfileerror</code> .                                                                                                                                                                           |
| <code>\listfiles</code>         | Giving this declaration in the preamble causes a list of all files input via the ‘safe input’ commands to be listed at the end. Any strings specified in the optional argument to <code>\ProvidesPackage</code> are listed alongside the file name. So files in standard (and other non-standard) distributions can put informative strings in this argument. |

## 4 Implementation

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | <code>1 &lt;*2ekernel&gt;</code>                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>\if@compatibility</code>   | The flag for compatibility mode.                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                                  | <code>2 \newif\if@compatibility</code>                                                                                                                                                                                                                                                                                                                                                                                                           |
|                                  | <i>(End of definition for \if@compatibility.)</i>                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>\@documentclasshook</code> | This legacy hook is called after the first <code>\documentclass</code> command. It is <i>not</i> integrated with the new 2020 hook management system! By default this checks to see if <code>\@normalsize</code> is undefined, and if so, sets it to <code>\normalsize</code> .                                                                                                                                                                  |
|                                  | <code>3 \def\@documentclasshook{% 4   \ifx\@normalsize\@undefined 5     \let\@normalsize\normalsize 6   \fi 7 }</code>                                                                                                                                                                                                                                                                                                                           |
|                                  | <i>(End of definition for \@documentclasshook.)</i>                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>\@declaredoptions</code>   | This list is automatically built by <code>\DeclareOption</code> . It is the list of options (separated by commas) declared in the class or package file and it defines the order in which the corresponding <code>\ds@\option</code> commands are executed. All local <code>\option</code> s which are not declared will be processed in the order defined by the optional argument of <code>\documentclass</code> or <code>\usepackage</code> . |
|                                  | <code>8 \let\@declaredoptions\empty</code>                                                                                                                                                                                                                                                                                                                                                                                                       |

(End of definition for \@declaredoptions.)

\@classoptionslist List of options of the main class.

```
 9 \let\@classoptionslist\relax
10 \%@\onlypreamble\@classoptionslist
```

(End of definition for \@classoptionslist.)

\@raw@classoptionslist List of options of the main class (unprocessed).

```
11 \let\@raw@classoptionslist\relax
```

(End of definition for \@raw@classoptionslist.)

\@unusedoptionlist List of options of the main class that haven't been declared or loaded as class option files.

```
12 \let\@unusedoptionlist\@empty
13 \%@\onlypreamble\@unusedoptionlist
```

(End of definition for \@unusedoptionlist.)

\CurrentOption Name of current package or option.

```
14 \let\CurrentOption\@empty
```

(End of definition for \CurrentOption.)

\@currpath Path to the current file if explicitly given.

```
15 </2ekernel>
16 <*2ekernel | latexrelease>
17 <latexrelease>
18 <latexrelease>\IncludeInRelease{2020/10/01}{\@currpath}%
19 <latexrelease> {Add \@currpath}%
20 \let\@currpath\@empty
21 <latexrelease>\EndIncludeInRelease
22 %
23 <latexrelease>\IncludeInRelease{0000/00/00}{\@currpath}%
24 <latexrelease> {Add \@currpath}%
25 <latexrelease>\let\@currpath\@undefined
26 <latexrelease>\EndIncludeInRelease
27 </2ekernel | latexrelease>
28 <*2ekernel>
```

(End of definition for \@currpath.)

\@currname Name of current package or option.

```
29 \let\@currname\@empty
```

(End of definition for \@currname.)

\@currext The current file extension.

```
30 \global\let\@currext=\@empty
```

(End of definition for \@currext.)

\@clsextension The two possible values of \@currext.

\@pkgextension  
31 \def\@clsextension{cls}  
32 \def\@pkgextension{sty}

(End of definition for \clsextension and \pkgextension.)

```
\@pushfilename Commands to push and pop the file name and extension.
\@popfilename #1 current name.
\currnamestack #2 current extension.
#3 current catcode of @.
#4 Rest of the stack.
33 </2ekernel>
34 {*2ekernel | latexrelease}
35 (latexrelease)
36 (latexrelease)\IncludeInRelease{2020/10/01}{\@pushfilename}%
37 (latexrelease) {Add \@expl@push@filename@@ and \@expl@push@filename@aux@@}%
38 \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).

```
39 \@expl@push@filename@@
40 \xdef\currnamestack{
41 {\@currname}%
42 {\@currext}%
43 {\the\catcode`@}%
44 \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.

```
45 \xdef\@kernel@currpathstack{
46 {\@currpath}%
47 \@kernel@currpathstack}%
48 \@expl@push@filename@aux@@}
49 (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.

```
50 (latexrelease)
51 (latexrelease)\IncludeInRelease{2020/02/02}{\@pushfilename}%
52 (latexrelease) {Add \@expl@push@filename@@}%
53 (latexrelease)\def\@pushfilename{
54 (latexrelease) \@expl@push@filename@@
55 (latexrelease) \xdef\currnamestack{
56 (latexrelease) {\@currname}%
57 (latexrelease) {\@currext}%
58 (latexrelease) {\the\catcode`@}%
59 (latexrelease) \currnamestack}%
60 (latexrelease) \@expl@push@filename@aux@@}
61 (latexrelease)\EndIncludeInRelease
62 (latexrelease)
```

When we roll back from a release that has `\expl3` preloaded, the definitions of `\@pushfilename` and `\@popfilename` can't be completely rolled back otherwise `\ExplSyntaxOff` based packages won't have the automatic `\ExplSyntaxOff` at the end. Here and below for `\@popfilename`, we don't roll back all the way through if coming from L<sup>A</sup>T<sub>E</sub>X > 2020 – 02 – 02.

```

63 \langle latexrelease\rangle\IncludeInRelease{0000/00/00}{\@pushfilename}%
64 \langle latexrelease\rangle {Add \Expl@push@filename@@ and \Expl@push@filename@aux@@}%
65 \langle latexrelease\rangle\ifnum\sourceLaTeXdate<20200202\relax
66 \langle latexrelease\rangle \GenericInfo{}{Defining 00-00-00\string\@pushfilename.}
67 \langle latexrelease\rangle\def\@pushfilename{%
68 \langle latexrelease\rangle \xdef\@currnamestack{%
69 \langle latexrelease\rangle {\@currname}%
70 \langle latexrelease\rangle {\@currext}%
71 \langle latexrelease\rangle {\the\catcode`\@}%
72 \langle latexrelease\rangle {\@currnamestack}%
73 \langle latexrelease\rangle\else
74 \langle latexrelease\rangle \GenericInfo{}{Defining 2020-02-02\string\@pushfilename.}
75 \langle latexrelease\rangle\def\@pushfilename{%
76 \langle latexrelease\rangle \Expl@push@filename@@
77 \langle latexrelease\rangle \xdef\@currnamestack{%
78 \langle latexrelease\rangle {\@currname}%
79 \langle latexrelease\rangle {\@currext}%
80 \langle latexrelease\rangle {\the\catcode`\@}%
81 \langle latexrelease\rangle {\@currnamestack}%
82 \langle latexrelease\rangle {\Expl@push@filename@aux@@}
83 \langle latexrelease\rangle\fi
84 \langle latexrelease\rangle\EndIncludeInRelease
85 \@onlypreamble\@pushfilename

86 \langle latexrelease\rangle
87 \langle latexrelease\rangle\IncludeInRelease{2020/10/01}{\@popfilename}%
88 \langle latexrelease\rangle {Add \Expl@pop@filename@@}%
89 \def\@popfilename{\Expl@@hook@curr@name@pop@@
90 \expandafter\@p@filename\@currnamestack\@nil

```

Same for popping:

```

91 \expandafter\@p@filepath\@kernel@currpathstack\@nil
92 \Expl@pop@filename@@
93 \langle latexrelease\rangle\EndIncludeInRelease
94 \langle latexrelease\rangle
95 \langle latexrelease\rangle\IncludeInRelease{2020/02/02}{\@popfilename}%
96 \langle latexrelease\rangle {Add \Expl@push@filename@@}%
97 \langle latexrelease\rangle\def\@popfilename{\expandafter\@p@filename\@currnamestack\@nil
98 \langle latexrelease\rangle {\Expl@pop@filename@@}
99 \langle latexrelease\rangle\EndIncludeInRelease
100 \langle latexrelease\rangle

101 \langle latexrelease\rangle\IncludeInRelease{0000/00/00}{\@popfilename}%
102 \langle latexrelease\rangle {Add \Expl@push@filename@@ and \Expl@push@filename@aux@@}%
103 \langle latexrelease\rangle\ifnum\sourceLaTeXdate<20200202\relax
104 \langle latexrelease\rangle \GenericInfo{}{Defining 00-00-00\string\@popfilename.}
105 \langle latexrelease\rangle\def\@popfilename{\expandafter\@p@filename\@currnamestack\@nil}
106 \langle latexrelease\rangle\else
107 \langle latexrelease\rangle \GenericInfo{}{Defining 2020-02-02\string\@popfilename.}

```

```

108 \def\@popfilename{\expandafter\@p@filename\@currnamestack\@nil
109 \def\@exp1\@pop@filename@@{%
110 \fi
111 \EndIncludeInRelease
112 \onlypreamble\@popfilename
113 {/2ekernel | latexrelease}
114 {*2ekernel}
115 \def\@p@filename#1#2#3#4\@nil{%
116 \gdef\@currname{#1}%
117 \gdef\@currext{#2}%
118 \catcode`\@#3\relax
119 \gdef\@currnamestack{#4}%
120 \onlypreamble\@p@filename
121 \gdef\@currnamestack{}%
122 \onlypreamble\@currnamestack

```

(End of definition for \@pushfilename, \@popfilename, and \@currnamestack.)

\@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.

```

123 {/2ekernel}
124 {*2ekernel | latexrelease}
125 \def\@currpathstack{%
126 \IncludeInRelease{2020/10/01}{\@kernel@\currpathstack}%
127 {Add \@kernel@\currpathstack}%

```

If rolling backwards to this release, \@kernel@\currpathstack will be defined, so the \gdef line should not be executed, thus the \@gobblethree will take it out, so the stack isn't touched.

```

128 \IfUndefined{\@kernel@\currpathstack}{}{\@gobblethree}
129 \gdef\@kernel@\currpathstack{}%

```

If rolling forward to this release, then the \gdef line above will define the path stack to be empty (which it can't be, inside a file), so the code below will traverse the \@currnamestack, and add as many empty items to \@kernel@\currpathstack as there are items in \@currnamestack, so both are back in sync. Most of the time latexrelease is loaded on top-level, so only one item is needed, but platexrelease loads it internally, so the more complicated loop is needed.

```

130 \ifx\@kernel@\currpathstack\empty
131 \def\reserved@a#1#2#3{%
132 \ifx\relax#3\else
133 \g@addto@macro\@kernel@\currpathstack{\{}%
134 \expandafter\reserved@a
135 \fi}%
136 \expandafter\reserved@a\@currnamestack{\{}{\relax}%
137 \fi
138 \def\@p@filepath#1{%
139 \gdef\@currpath{#1}\@p@filepath@aux\empty}
140 \def\@p@filepath@aux#1\@nil{%
141 \xdef\@kernel@\currpathstack{#1}}
142 \EndIncludeInRelease
143 %
144 \IncludeInRelease{0000/00/00}{\@kernel@\currpathstack}%

```

```

145 〈latexrelease〉 {Add \@kernel@currpathstack}%
146 〈latexrelease〉\let\@kernel@currpathstack\@undefined
147 〈latexrelease〉\let\@p@filepath\@undefined
148 〈latexrelease〉\let\@p@filepath@aux\@undefined
149 〈latexrelease〉\EndIncludeInRelease
150 〈/2ekernel | latexrelease〉
151 〈*2ekernel〉

```

(End of definition for \@kernel@currpathstack.)

\@optionlist Returns the option list of the file.

```

152 \def\@optionlist#1{%
153 \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}%
154 }%\@onlypreamble\@optionlist

```

(End of definition for \@optionlist.)

\@ifpackageloaded \@ifpackageloaded{⟨name⟩} Checks to see whether a file has been loaded.

```

155 \def\@ifpackageloaded{\@ifl@aded\@pkgextension}
156 \def\@ifclassloaded{\@ifl@aded\@clsextension}
157 \def\@ifl@aded#1#2{%
158 \expandafter\ifx\csname ver@#2.#1\endcsname\relax
159 \expandafter\@secondoftwo
160 \else
161 \expandafter\@firstoftwo
162 \fi}

```

(End of definition for \@ifpackageloaded and \@ifclassloaded.)

\@ifpackagelater \@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 be \@ifpackagelaterorequal but it is in use for more than 30 years, so ...

```

163 \def\@ifpackagelater{\@ifl@ter\@pkgextension}
164 \def\@ifclasslater{\@ifl@ter\@clsextension}

```

(End of definition for \@ifpackagelater and \@ifclasslater.)

\IfPackageAtLeastTF \IfFormatAtLeastTF{YYYY/MM/DD}{⟨true code⟩}{⟨false code⟩} Test if the format is later or equal to the given date.

\IfClassAtLeastTF 165 〈/2ekernel〉

\IfFileAtLeastTF 166 〈\*2ekernel | latexrelease〉

```

167 〈latexrelease〉\IncludeInRelease{2020/10/01}%
168 〈latexrelease〉 {\IfFormatAtLeastTF}{Test format date}%
169 \def\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
170 \let\IfPackageAtLeastTF\@ifpackagelater
171 \let\IfClassAtLeastTF\@ifclasslater
172 \def\IfFileAtLeastTF#1{\expandafter\@ifl@t@r\csname ver@#1\endcsname}

```

For rollback pretend it was available since the beginning of dawn.

```

173 〈/2ekernel | latexrelease〉
174 〈latexrelease〉\EndIncludeInRelease
175 〈latexrelease〉\IncludeInRelease{0000/00/00}%
176 〈latexrelease〉 {\IfFormatAtLeastTF}{Test format date}%
177 〈latexrelease〉\def\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}

```

```

178 \let\IfPackageAtLeastTF\@ifpackagelater
179 \let\IfClassAtLeastTF\@ifclasslater
180 \def\IfFileAtLeastTF#1{\expandafter\@ifl@t@r\csname ver@#1\endcsname}
181 \EndIncludeInRelease
182 {*2ekernel}

```

(*End of definition for \IfPackageAtLeastTF and others.*)

\@ifl@ter

```

183 \def\@ifl@ter#1#2{%
184 \expandafter\@ifl@t@r
185 \csname ver@#2.#1\endcsname
186 {*}2ekernel}

This internal macro is also used in \NeedsTeXFormat.

187 \IncludeInRelease[2018/04/01]%
188 { \@ifl@t@r}{Guard against bad input}%
189 {*2ekernel | latexrelease}
190 \def\@ifl@t@r#1#2{%
191 \ifnum\expandafter\@parse@version@#1//00\@nil<%
192 \expandafter\@parse@version@#2//00\@nil
193 \expandafter\@secondoftwo
194 \else
195 \expandafter\@firstoftwo
196 \fi}
197 \def\@parse@version@#1{\@parse@version0#1}
198 {*}2ekernel | latexrelease}
199 \EndIncludeInRelease
200 \IncludeInRelease[0000/00/00]%
201 { \@ifl@t@r}{Guard against bad input}%
202 \def\@ifl@t@r#1#2{%
203 \ifnum\expandafter\@parse@version#1//00\@nil<%
204 \expandafter\@parse@version#2//00\@nil
205 \expandafter\@secondoftwo
206 \else
207 \expandafter\@firstoftwo
208 \fi}
209 \let\@parse@version@\undefined
210 \EndIncludeInRelease
211 {*}2ekernel}

```

(*End of definition for \@ifl@ter.*)

```

212 {*}2ekernel}
213 {*2ekernel | latexreleasefirst}
214 \def\@parse@version#1/#2/#3#4#5\@nil{%
215 \@parse@version@dash#1-#2-#3#4\@nil
216 }

```

The \if test here ensures that an argument with no / or - produces 0 (actually 00).

```

217 \def\@parse@version@dash#1-#2-#3#4#5\@nil{%
218 \if\relax#2\relax\else#1\fi#2#3#4 }
219 {*}2ekernel | latexreleasefirst}
220 {*}2ekernel}

```

```

\@ifpackagewith \@ifpackagewith{\langle name\rangle}{\langle option-list\rangle} Checks that \langle option-list\rangle is a subset of
\@ifclasswith the options with which \langle name\rangle was loaded.

221 \def\@ifpackagewith{\@ifoptions\@pkgextension}
222 \def\@ifclasswith{\@ifoptions\@clsextension}
223 \def\@ifoptions#1#2{%
224 \@expandtwoargs\@ifoptions{\@optionlist{#2.#1}}}

Probably shouldn't use \CurrentOption here... (changed to \reserved@b.)

225 </2ekernel>
226 <latexrelease>\IncludeInRelease{2017/01/01}%
227 <latexrelease> {\@ifoptions{Spaces in option clash check}%
228 <*2ekernel | latexrelease>
229 \def\@ifoptions#1#2{%
230 \let\reserved@a\@firstoftwo

231 \edef\reserved@b{\zap@space#2 \empty}%
232 \@for\reserved@b:=\reserved@b\do{%
233 \ifx\reserved@b\empty
234 \else
235 \expandafter\in@\expandafter{\expandafter,\reserved@b,}{,#1,}%
236 \ifin@
237 \else
238 \let\reserved@a\@secondoftwo
239 \fi
240 \fi
241 }%
242 \reserved@a
243 </2ekernel | latexrelease>
244 <latexrelease>\EndIncludeInRelease
245 <latexrelease>\IncludeInRelease{0000/00/00}%
246 <latexrelease> {\@ifoptions{Spaces in option clash check}%
247 <latexrelease>\def\@ifoptions#1#2{%
248 <latexrelease> \let\reserved@a\@firstoftwo
249 <latexrelease> \@for\reserved@b:=#2\do{%
250 <latexrelease> \ifx\reserved@b\empty
251 <latexrelease> \else
252 <latexrelease> \expandafter\in@\expandafter
253 {\expandafter,\reserved@b,}{,#1,}%
254 <latexrelease> \ifin@
255 <latexrelease> \else
256 <latexrelease> \let\reserved@a\@secondoftwo
257 <latexrelease> \fi
258 <latexrelease> \fi
259 }%
260 <latexrelease> \reserved@a
261 <latexrelease>\EndIncludeInRelease
262 <*2ekernel>

```

(End of definition for \@ifpackagewith and \@ifclasswith.)

|                               |                                                                       |
|-------------------------------|-----------------------------------------------------------------------|
| \IfPackageLoadedTF            | More public names for the commands already available for a long time. |
| \IfPackageLoadedWithOptionsTF |                                                                       |
| \IfClassLoadedTF              |                                                                       |
| \IfClassLoadedWithOptionsTF   |                                                                       |

```

263 </2ekernel>
264 <*2ekernel | latexrelease>
265 <latexrelease>\IncludeInRelease{2024/06/01}%

```

```

266 \let \IfPackageLoadedTF {\IfPackageLoadedTF}{Test package loading}%
267 \let \IfClassLoadedTF {\IfClassLoadedTF}%
268 \let \IfPackageLoadedWithOptionsTF {\IfPackageLoadedWithOptionsTF}%
269 \let \IfClassLoadedWithOptionsTF {\IfClassLoadedWithOptionsTF}%
270

```

For rollback/rollforward pretend everything was available since the beginning of dawn.

```

271 \let \IfPackageLoadedTF {\IfPackageLoadedTF}%
272 \let \IfClassLoadedTF {\IfClassLoadedTF}%
273 \let \IfPackageLoadedWithOptionsTF {\IfPackageLoadedWithOptionsTF}%
274 \let \IfClassLoadedWithOptionsTF {\IfClassLoadedWithOptionsTF}%
275
276 \let \IfPackageLoadedTF {\IfPackageLoadedTF}%
277 \let \IfClassLoadedTF {\IfClassLoadedTF}%
278 \let \IfPackageLoadedWithOptionsTF {\IfPackageLoadedWithOptionsTF}%
279 \let \IfClassLoadedWithOptionsTF {\IfClassLoadedWithOptionsTF}%
280
281 \let \IfPackageLoadedTF {\IfPackageLoadedTF}%
282 \let \IfClassLoadedTF {\IfClassLoadedTF}%
283

```

(End of definition for \IfPackageLoadedTF and others.)

A few more conditionals for convenience

```

283 \def \IfPackageAtLeastT {\IfPackageAtLeastT}%
284 \def \IfPackageAtLeastF {\IfPackageAtLeastF}%
285 \def \IfClassAtLeastT {\IfClassAtLeastT}%
286 \def \IfClassAtLeastF {\IfClassAtLeastF}%
287 \def \IfFileAtLeastT {\IfFileAtLeastT}%
288 \def \IfFileAtLeastF {\IfFileAtLeastF}%
289 \def \IfFormatAtLeastT {\IfFormatAtLeastT}%
290 \def \IfFormatAtLeastF {\IfFormatAtLeastF}%
291 \def \IfPackageLoadedWithOptionsT {\IfPackageLoadedWithOptionsT}%
292 \def \IfPackageLoadedWithOptionsF {\IfPackageLoadedWithOptionsF}%
293 \def \IfClassLoadedT {\IfClassLoadedT}%
294 \def \IfClassLoadedF {\IfClassLoadedF}%
295 \def \IfFileAtLeastT {\IfFileAtLeastT}%
296 \def \IfFileAtLeastF {\IfFileAtLeastF}%
297 \def \IfFormatAtLeastT {\IfFormatAtLeastT}%
298 \def \IfFormatAtLeastF {\IfFormatAtLeastF}%
299 \def \IfPackageLoadedWithOptionsTF {\IfPackageLoadedWithOptionsTF}%
300 \def \IfPackageLoadedWithOptionsF {\IfPackageLoadedWithOptionsF}%
301 \def \IfClassLoadedWithOptionsT {\IfClassLoadedWithOptionsT}%
302 \def \IfClassLoadedWithOptionsF {\IfClassLoadedWithOptionsF}%

```

These three commands haven't been there at all in the past.

```

303 \def \IfFileLoadedTF{\%
304 \expandafter\ifx\csname ver@#1\endcsname\relax
305 \expandafter\@secondoftwo
306 \else
307 \expandafter\@firstoftwo
308 \fi}
309 \def \IfFileLoadedT{\#1\#2{\IfFileLoadedTF{\#1}{\#2}}}
310 \def \IfFileLoadedF {\#1{\IfFileLoadedTF{\#1}}}

```

For rollback/rollforward pretend everything was available since the beginning of dawn.

```

311 </2ekernel | latexrelease>
312 <latexrelease>\EndIncludeInRelease
313 <latexrelease>\IncludeInRelease{0000/00/00}%
314 <latexrelease> {\IfPackageLoadedT}{More conditionals}%
315 <latexrelease>
316 <latexrelease>\def\IfPackageLoadedT #1#2{\IfPackageLoadedTF{#1}{#2}{}}%
317 <latexrelease>\def\IfPackageLoadedF #1{\IfPackageLoadedTF{#1}{}}%
318 <latexrelease>\def\IfClassLoadedT #1#2{\IfClassLoadedTF{#1}{#2}{}}%
319 <latexrelease>\def\IfClassLoadedF #1{\IfClassLoadedTF{#1}{}}%
320 <latexrelease>\def\IfPackageAtLeastT#1#2#3{\IfPackageAtLeastTF{#1}{#2}{#3}{}}%
321 <latexrelease>\def\IfPackageAtLeastF #1#2{\IfPackageAtLeastTF{#1}{#2}{}}%
322 <latexrelease>\def\IfClassAtLeastT #1#2#3{\IfClassAtLeastTF{#1}{#2}{#3}{}}%
323 <latexrelease>\def\IfClassAtLeastF #1#2{\IfClassAtLeastTF{#1}{#2}{}}%
324 <latexrelease>\def\IfFileAtLeastT #1#2#3{\IfFileAtLeastTF{#1}{#2}{#3}{}}%
325 <latexrelease>\def\IfFileAtLeastF #1#2{\IfFileAtLeastTF{#1}{#2}{}}%
326 <latexrelease>\def\IfFormatAtLeastT #1#2{\IfFormatAtLeastTF{#1}{#2}{}}%
327 <latexrelease>\def\IfFormatAtLeastF #1{\IfFormatAtLeastTF{#1}{}}%
328 <latexrelease>\def\IfPackageLoadedWithOptionsT #1#2#3{\IfPackageLoadedWithOptionsTF{#1}{#2}{#3}{}}%
329 <latexrelease>\def\IfPackageLoadedWithOptionsF #1#2{\IfPackageLoadedWithOptionsTF{#1}{#2}{}}%
330 <latexrelease>\def\IfClassLoadedWithOptionsT #1#2#3{\IfClassLoadedWithOptionsTF{#1}{#2}{#3}{}}%
331 <latexrelease>\def\IfClassLoadedWithOptionsF #1#2{\IfClassLoadedWithOptionsTF{#1}{#2}{}}%
332 <latexrelease>
333 <latexrelease>\def\IfFileLoadedTF#1{%
334 <latexrelease> \expandafter\ifx\csname ver@#1\endcsname\relax
335 <latexrelease> \expandafter\@secondoftwo
336 <latexrelease> \else
337 <latexrelease> \expandafter\@firstoftwo
338 <latexrelease> \fi}
339 <latexrelease>\def\IfFileLoadedT#1#2{\IfFileLoadedTF{#1}{#2}{}}%
340 <latexrelease>\def\IfFileLoadedF #1{\IfFileLoadedTF{#1}{}}%
341 <latexrelease>
342 <latexrelease>\EndIncludeInRelease
343 <*2ekernel>

```

(End of definition for \IfPackageLoadedT and others.)

\ProvidesPackage Checks that the current filename is correct, and defines \ver@filename.

```

344 </2ekernel>
345 <latexrelease>\IncludeInRelease{2020/10/01}%
346 <latexrelease> {\ProvidesPackage}{Check name with \strcmp}%
347 <*2ekernel | latexrelease>
348 \def\ProvidesPackage#1{%
349 \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.

```

350 \@expandtwoargs\@expl@str@if@eq@nnTF
351 {\@gtempa}{\@currpath\@currname}{}{%
352 \@latex@warning@no@line{You have requested
353 \@cls@pkg\space`@\currpath\@currname',\MessageBreak
354 but the \@cls@pkg\space provides '#1'}%
355 }%

```

```

356 \@ifnextchar[\@pr@videopackage{\@pr@videopackage[]}]%
357 \onlypreamble\ProvidesPackage
358 </2ekernel | latexrelease>
359 <latexrelease>\EndIncludeInRelease
360 %
361 <latexrelease>\IncludeInRelease{0000/00/00}%
362 <latexrelease> {\ProvidesPackage}{Undo: check name with \strcmp}%
363 <latexrelease>\def\ProvidesPackage#1{%
364 <latexrelease> \xdef\@gtempa{#1}%
365 <latexrelease> \ifx\@gtempa\currname\else
366 <latexrelease> \@latex@warning@no@line{You have requested
367 <latexrelease> \@cls@pkg\space`\currname', \MessageBreak
368 <latexrelease> but the \@cls@pkg\space provides '#1'}%
369 <latexrelease> \fi
370 <latexrelease> \ifnextchar[\@pr@videopackage{\@pr@videopackage[]}]%
371 <latexrelease>\EndIncludeInRelease
372 <2ekernel>

```

(End of definition for `\ProvidesPackage`.)

`\@pr@videopackage` This is the helper command for `\ProvidesPackage`. It tries to be cautious when handling the identification string in case it contains UTF-8 characters.

```

373 </2ekernel>
374 <2ekernel | latexrelease>
375 <latexrelease>\IncludeInRelease{2020/10/01}%
376 <latexrelease> {\@pr@videopackage}{Allow for package substitution}%
377 \def\@pr@videopackage[#1]{%
378 \expandafter\protected@\xdef % <-- protected...
379 \csname ver@\currname.\@currext\endcsname{#1}% Loaded package
380 \expandafter\let
381 \csname ver@\currpkg@reqd\expandafter\endcsname % Requested package
382 \csname ver@\currname.\@currext\endcsname
383 \ifx\@currext\clsextension
384 \typeout{Document Class: \@gtempa\space#1}%
385 \else
386 \protected@wlog{Package: \@gtempa\space#1}% <--- protected
387 \fi}

```

(End of definition for `\@pr@videopackage`.)

`\protected@wlog` This is like plain TeX's `\wlog` but gracefully handles protected commands.

```

388 \long\def\protected@wlog#1{\begingroup
389 \set@display@protect
390 \immediate \write \m@ne {#1}\endgroup }

```

(End of definition for `\protected@wlog`.)

```

391 </2ekernel | latexrelease>
392 <latexrelease>\EndIncludeInRelease
393 <latexrelease>\IncludeInRelease{2020/02/02}%
394 <latexrelease> {\@pr@videopackage}{Protection for package info}%
395 <latexrelease>
396 <latexrelease>\def\@pr@videopackage[#1]{%
397 <latexrelease> \expandafter\protected@\xdef % <-- protected...
398 <latexrelease> \csname ver@\currname.\@currext\endcsname{#1}%

```

```

399 \ifx\@currext\@clsextension
400 \typeout{Document Class: \@gtempa\space#1}%
401 \else
402 \protected@wlog{Package: \@gtempa\space#1}%
403 \fi}
404 \else
405 \EndIncludeInRelease
406 \IncludeInRelease{0000/00/00}%
407 {\@pr@videopackage}{Protection for package info}%
408 \else
409 \def\@pr@videopackage[#1]{%
410 \expandafter\xdef\csname ver@\@currname.\@currext\endcsname{#1}%
411 \ifx\@currext\@clsextension
412 \typeout{Document Class: \@gtempa\space#1}%
413 \else
414 \wlog{Package: \@gtempa\space#1}%
415 \fi}
416 \let\protected@wlog\@undefined
417 \else
418 \EndIncludeInRelease
419 \else
420 \onlypreamble\@pr@videopackage

```

**\ProvidesClass** Like `\ProvidesPackage`, but for classes. This needs a dummy `\textralase` block to copy the definition of `\ProvidesPackage` as it changes across releases.

```

421 \else
422 \IncludeInRelease{0000/00/00}%
423 {\@ProvidesClass}{Track \ProvidesPackage}%
424 \else
425 \let\ProvidesClass\ProvidesPackage
426 \onlypreamble\ProvidesClass
427 \else
428 \EndIncludeInRelease
429 \else

```

(*End of definition for \ProvidesClass.*)

**\ProvidesFile** Like `\ProvidesPackage`, but for arbitrary files. Do not apply `\onlypreamble` to these, as we may want to label files input during the document.

```

\@providesfile 430 \def\ProvidesFile#1{%
431 \begingroup
432 \catcode`\ 10 %
433 \ifnum \endlinechar<256 %
434 \ifnum \endlinechar>\m@ne
435 \catcode\endlinechar 10 %
436 \fi
437 \fi
438 \makeother\%
439 \makeother\&%
440 \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 `ltffinal.dtx`.

```
def\@providesfile#1[#2]{%
 \wlog{File: #1 #2}%
 \expandafter\xdef\csname ver@#1\endcsname{#2}%
 \endgroup}
(End of definition for \ProvidesFile and \@providesfile.)
```

\PassOptionsToPackage If the package has been loaded, we check that it was first loaded with the options.  
 \PassOptionsToClass Otherwise we add the option list to that of the package.

```
441 </2ekernel>
442 <latexrelease>\IncludeInRelease{2021/06/01}%
443 <latexrelease> {\@pass@ptions}{Raw option lists}%
444 <2ekernel | latexrelease>
445 \def\@pass@ptions#1#2#3{%
446 \@expl@@@filehook@set@curr@file@@nN
447 {\@expl@@@filehook@resolve@file@subst@w #3.#1\@nil}%
448 \reserved@a\reserved@b
449 \@expl@@@filehook@clear@replacement@flag@@
450 \expandafter\protected\xdef\csname opt@\reserved@a\endcsname{%
451 \@ifundefined{opt@\reserved@a}\@empty
452 {\csname opt@\reserved@a\endcsname,}%
453 \zap@space#2 \@empty}%
454 \expandafter\let
455 \csname opt@#3.#1\expandafter\endcsname
456 \csname opt@\reserved@a\endcsname
```

Extend raw option list

```
457 \@ifundefined{@raw@opt@#3.#1}%
458 {\expandafter\gdef\csname @raw@opt@#3.#1\expandafter\endcsname
459 \expandafter{\#2}%
460 {\expandafter\g@addto@macro\csname @raw@opt@#3.#1\expandafter\endcsname
461 \expandafter{\expandafter,\#2}}%
462 }
463 </2ekernel | latexrelease>
464 <latexrelease>\EndIncludeInRelease
465 <latexrelease>\IncludeInRelease{2020/10/01}{\@pass@ptions}
466 <latexrelease> {Add file replacement in \@pass@ptions}%
467 <latexrelease>
468 <latexrelease>\def\@pass@ptions#1#2#3{%
469 <latexrelease> \@expl@@@filehook@set@curr@file@@nN
470 <latexrelease> {\@expl@@@filehook@resolve@file@subst@w #3.#1\@nil}%
471 <latexrelease> \reserved@a\reserved@b
472 <latexrelease> \@expl@@@filehook@clear@replacement@flag@@
473 <latexrelease> \expandafter\xdef\csname opt@\reserved@a\endcsname{%
474 <latexrelease> \@ifundefined{opt@\reserved@a}\@empty
475 <latexrelease> {\csname opt@\reserved@a\endcsname,}%
476 <latexrelease> \zap@space#2 \@empty}%
477 <latexrelease> \expandafter\let
478 <latexrelease> \csname opt@#3.#1\expandafter\endcsname
479 <latexrelease> \csname opt@\reserved@a\endcsname}
480 <latexrelease>\EndIncludeInRelease
```

```

481 〈\latexrelease〉\IncludeInRelease{0000/00/00}{\@pass@ptions}
482 〈\latexrelease〉 {\@pass@ptions}%
483 〈\latexrelease〉
484 〈\latexrelease〉\def\@pass@ptions#1#2#3{%
485 〈\latexrelease〉 \expandafter\xdef\csname opt@#3.#1\endcsname{%
486 〈\latexrelease〉 \@ifundefined{opt@#3.#1}\@empty
487 〈\latexrelease〉 {\csname opt@#3.#1\endcsname,}%
488 〈\latexrelease〉 \zap@space#2 \@empty}%
489 〈\latexrelease〉\EndIncludeInRelease
490 {*2ekernel}
491 \@onlypreamble\@pass@ptions
492 \def\PassOptionsToPackage{\@pass@ptions\@pkgextension}
493 \def\PassOptionsToClass{\@pass@ptions\@clsextension}
494 \@onlypreamble\PassOptionsToPackage
495 \@onlypreamble\PassOptionsToClass

```

(End of definition for *\PassOptionsToPackage* and *\PassOptionsToClass*.)

**\DeclareOption** Adds an option as a *\ds@* command, or the default *\default@ds* command.

**\DeclareOption\***

```

496 \def\DeclareOption{%
497 \let\@fileswith@ptions\@badrequireerror
498 \@ifstar\@defdefault@ds\@declareoption}
499 \long\def\@declareoption#1#2{%
500 \xdef\@declaredoptions{\@declaredoptions,#1}%
501 \toks@{#2}%
502 \expandafter\edef\csname ds@#1\endcsname{\the\toks@}%
503 \long\def\@defdefault@ds#1{%
504 \toks@{#1}%
505 \edef\default@ds{\the\toks@}%
506 \@onlypreamble\DeclareOption
507 \@onlypreamble\@declareoption
508 \@onlypreamble\@defdefault@ds

```

(End of definition for *\DeclareOption* and *\DeclareOption\**.)

**\OptionNotUsed** If we are in a class file, add *\CurrentOption* to the list of unused options. Otherwise, in a package file do nothing.

```

509 〈/2ekernel〉
510 〈\latexrelease〉\IncludeInRelease{2021/06/01}%
511 〈\latexrelease〉 {\@OptionNotUsed}{filter unused option list}%
512 {*2ekernel | latexrelease}
513 \ExplSyntaxOn
514 \def\@remove@eq@value#1=#2\@nil{\tl_trim_spaces:n{#1}}
515 \ExplSyntaxOff
516 \def\@OptionNotUsed{%
517 \ifx\@curr ext\@clsextension
518 \xdef\@unusedoptionlist{%
519 \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
520 \expandafter\@remove@eq@value\CurrentOption=\@nil}%
521 \fi}
522 〈/2ekernel | latexrelease〉
523 〈\latexrelease〉\EndIncludeInRelease
524 〈\latexrelease〉\IncludeInRelease{0000/00/00}%

```

```

525 〈latexrelease〉 {\OptionNotUsed}{filter unused option list}%
526 〈latexrelease〉\let\@remove@eq@value\@undefined
527 〈latexrelease〉\def\OptionNotUsed{%
528 〈latexrelease〉 \ifx\@current@clsextension
529 〈latexrelease〉 \xdef\@unusedoptionlist{%
530 〈latexrelease〉 \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
531 〈latexrelease〉 \CurrentOption}%
532 〈latexrelease〉 \fi}
533 〈latexrelease〉\EndIncludeInRelease
534 {*2ekernel}
535 \onlypreamble\OptionNotUsed

```

(End of definition for `\OptionNotUsed` and `\@remove@eq@value`.)

`\default@ds` The default option code. Set by `\onefilewithoptions` to either `\OptionNotUsed` for classes, or `\unknownonerror` for packages. This may be reset in either case with `\DeclareOption*`.

```
536 % \let\default@ds\OptionNotUsed
```

(End of definition for `\default@ds`.)

`\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.

```

537 \def\ProcessOptions{%
538 \let\ds@\empty
539 \protected\edef\@curroptions{\optionlist{\currname.\@current} }%
540 \@ifstar\xprocessoptions\processoptions
541 \onlypreamble\ProcessOptions

542 \def\@processoptions{%
543 \@for\CurrentOption:=\@declaredoptions\do{%
544 \ifx\CurrentOption\empty\else
545 \@expandtwoargs\in@\,\CurrentOption,\}{}%
546 ,\ifx\@current\@clsextension\else\@classoptionslist,\fi
547 \@curroptions,\}%
548 \ifin@
549 \useoption
550 \expandafter\let\csname ds@\CurrentOption\endcsname\empty
551 \fi
552 \fi}%
553 \processoptions
554 \onlypreamble\processoptions

555 {*}2ekernel}
556 〈latexrelease〉\IncludeInRelease{2021/06/01}%
557 〈latexrelease〉 {\@xprocessoptions}{safer @xprocessoptions}%
558 {*2ekernel | latexrelease}
559 \def\@xprocessoptions{%

```

```

560 \ifx\@currrent\@clsextension\else
561 \ifx\@classoptionslist\relax\else
562 \@for\CurrentOption:=\@classoptionslist\do{%
563 \ifx\CurrentOption\@empty\else
564 \@ifundefined{ds@\detokenize\expandafter{\CurrentOption}}{}{%
565 \use@option
566 \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
567 }%
568 \fi}%
569 \fi
570 \fi
571 \@process@pti@ns}
572 {/2ekernel | latexrelease}
573 {latexrelease}\EndIncludeInRelease
574 {latexrelease}\IncludeInRelease{0000/00/00}%
575 {latexrelease} {\@xprocess@ptions}{safer @xprocess@ptions}%
576 {latexrelease}\let\@remove@eq@value\@undefined
577 {latexrelease}\def\@xprocess@ptions{%
578 {latexrelease} \ifx\@currrent\@clsextension\else
579 {latexrelease} \@for\CurrentOption:=\@classoptionslist\do{%
580 {latexrelease} \ifx\CurrentOption\@empty\else
581 {latexrelease} \@expandtwoargs\in@{\CurrentOption,}{,\@declaredoptions,}%
582 {latexrelease} \ifin@
583 {latexrelease} \use@option
584 {latexrelease} \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
585 {latexrelease} \fi
586 {latexrelease} \fi}%
587 {latexrelease} \fi
588 {latexrelease} \@process@pti@ns}
589 {latexrelease}\EndIncludeInRelease
590 {*2ekernel}
591 \onlypreamble\@xprocess@ptions

The common part of \ProcessOptions and \ProcessOptions*.

592 {/2ekernel}
593 {*2ekernel | latexrelease}
594 {latexrelease}\IncludeInRelease{2020/10/01}%
595 {latexrelease} {\@process@pti@ns}{Unused options issue}%
596 \def\@process@pti@ns{%
597 \@for\CurrentOption:=\@curroptions\do{%
598 \@ifundefined{ds@\detokenize\expandafter{\CurrentOption}}{%
599 \use@option
600 \default@ds}}%
601 \use@option}%

```

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`.

Clear all the definitions for option code. First set all the declared options to `\relax`, then reset the ‘default’ and ‘empty’ options. and the list of declared options.

```

602 \@for\CurrentOption:=\@declaredoptions\do{%
603 \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%

```

```

604 \let\CurrentOption\@empty
605 \let\@fileswith@pti@ns\@@fileswith@pti@ns
606 \AtEndOfPackage{\expandafter\let
607 \csname unprocessedoptions-\@currname.\@currext\endcsname
608 \relax}
609 \@onlypreamble\@process@pti@ns
610 </2ekernel | latexrelease>
611 <latexrelease>\EndIncludeInRelease
612 <latexrelease>\IncludeInRelease{0000/00/00}%
613 <latexrelease> {\@process@pti@ns}{Unused options issue}%
614 <latexrelease>
615 <latexrelease>\def\@process@pti@ns{%
616 <latexrelease> \@for\CurrentOption:=\@curroptions\do{%
617 <latexrelease> \@ifundefined{ds@\CurrentOption}%
618 <latexrelease> {\@use@option
619 <latexrelease> \default@ds}%
620 <latexrelease> \@use@option}%
621 <latexrelease> \@for\CurrentOption:=\@declaredoptions\do{%
622 <latexrelease> \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%
623 <latexrelease> \let\CurrentOption\@empty
624 <latexrelease> \let\@fileswith@pti@ns\@@fileswith@pti@ns
625 <latexrelease> \AtEndOfPackage{\let\@unprocessedoptions\relax}
626 <latexrelease>\EndIncludeInRelease
627 <*2ekernel>

```

(End of definition for \ProcessOptions and \ProcessOptions\*.)

\@options \@options is a synonym for \ProcessOptions\* for upward compatibility with L<sup>A</sup>T<sub>E</sub>X2.09 style files.

```

628 \def\@options{\ProcessOptions*}
629 \@onlypreamble\@options

```

(End of definition for \@options.)

\@use@option Execute the code for the current option.

```

630 </2ekernel>
631 <latexrelease>\IncludeInRelease{2021/06/01}%
632 <latexrelease> {\@use@option}{filter unused option list}%
633 <*2ekernel | latexrelease>
634 \def\@use@option{%
635 \@expandtwoargs\@removeelement
636 {\expandafter\@remove@eq@value\CurrentOption=\@nil}%
637 \@unusedoptionlist\@unusedoptionlist
638 \csname ds@\detokenize\expandafter{\CurrentOption}\endcsname
639 </2ekernel | latexrelease>
640 <latexrelease>\EndIncludeInRelease
641 <latexrelease>\IncludeInRelease{0000/00/00}%
642 <latexrelease> {\@use@option}{filter unused option list}%
643 <latexrelease>\def\@use@option{%
644 \@expandtwoargs\@removeelement\CurrentOption
645 \@unusedoptionlist\@unusedoptionlist
646 \csname ds@\CurrentOption\endcsname
647 <latexrelease>\EndIncludeInRelease
648 <*2ekernel>

```

```
649 \onlypreamble\useoption
```

(End of definition for \useoption.)

\ExecuteOptions \ExecuteOptions{\option-list} executes the code declared for each option.

```
650 </2ekernel>
651 <latexrelease>\IncludeInRelease{2017/01/01}%
652 <latexrelease> {\ExecuteOptions}{Spaces in \ExecuteOptions}%
653 <*2ekernel | latexrelease>
654 \def\ExecuteOptions#1{%
```

Use \@fortmp here as it is anyway cleared during \for loop so does not change any existing names.

```
655 \edef\@formp{\zap@space#1 \@empty}%
656 \def\reserved@a##1@nil{%
657 \@for\CurrentOption:=\@formp\do
658 {\csname ds@\CurrentOption\endcsname}%
659 \edef\CurrentOption{##1}%
660 \expandafter\reserved@a\CurrentOption@nil}
661 </2ekernel | latexrelease>
662 <latexrelease>\EndIncludeInRelease
663 <latexrelease>\IncludeInRelease{0000/00/00}%
664 <latexrelease> {\ExecuteOptions}{Spaces in \ExecuteOptions}%
665 <latexrelease>\def\ExecuteOptions#1{%
666 \def\reserved@a##1@nil{%
667 \@for\CurrentOption:=#1\do
668 {\csname ds@\CurrentOption\endcsname}%
669 \edef\CurrentOption{##1}%
670 \expandafter\reserved@a\CurrentOption@nil}
671 <latexrelease>\EndIncludeInRelease
672 <*2ekernel>
673 \onlypreamble\ExecuteOptions
```

(End of definition for \ExecuteOptions.)

The top-level commands, which just set some parameters then call the internal command, \@fileswoptions.

\documentclass The main new-style class declaration.

```
674 \def\documentclass{%
675 \let\documentclass\@twoclasseserror
676 \if@compatibility\else\let\usepackage\RequirePackage\fi
677 \@fileswoptions\@clsextension}
678 \onlypreamble\documentclass
```

(End of definition for \documentclass.)

\documentstyle 2.09 style class ‘style’ declaration.

```
679 \def\documentstyle{%
680 \makeatletter\input{latex209.def}\makeatother
681 \documentclass}
682 \onlypreamble\documentstyle
```

(End of definition for \documentstyle.)

\RequirePackage Load package if not already loaded.

```
683 \def\RequirePackage{%
684 \@fileswithoptions\@pkgextension}
685 \onlypreamble\RequirePackage
```

(End of definition for \RequirePackage.)

\LoadClass Load class.

```
686 \def\LoadClass{%
687 \ifx\@currext\@pkgextension
688 \@latex@error
689 {\noexpand\LoadClass in package file}%
690 {You may only use \noexpand\LoadClass in a class file.}%
691 \fi
692 \@fileswithoptions\@clsextension}
693 \onlypreamble\LoadClass
```

(End of definition for \LoadClass.)

\@loadwithoptions Pass the current option list on to a class or package. #1 is \@cls-or-pkgextension, #2 is \RequirePackage or \LoadClass, #3 is the class or package to be loaded.

```
694 </2ekernel>
695 <latexrelease>\IncludeInRelease{2021/06/01}%
696 <latexrelease> {\@loadwithoptions}{Raw option lists load with options}%
697 <*2ekernel | latexrelease>
698 \def\@loadwithoptions#1#2#3{%
699 \expandafter\let\csname opt@\#3.\#1\expandafter\endcsname
700 \csname opt@\@currname.\@currext\endcsname
701 \expandafter\let\csname @raw@opt@\#3.\#1\expandafter\endcsname
702 \csname @raw@opt@\@currname.\@currext\endcsname
703 #2{\#3}}
704 </2ekernel | latexrelease>
705 <latexrelease>\EndIncludeInRelease
706 <latexrelease>\IncludeInRelease{0000/00/00}
707 <latexrelease> {\@loadwithoptions}{Raw option lists load with options}%
708 <latexrelease>\def\@loadwithoptions#1#2#3{%
709 \expandafter\let\csname opt@\#3.\#1\expandafter\endcsname
710 \csname opt@\@currname.\@currext\endcsname
711 #2{\#3}}
712 <latexrelease>\EndIncludeInRelease
713 <*2ekernel>
714 \onlypreamble\@loadwithoptions
```

(End of definition for \@loadwithoptions.)

\LoadClassWithOptions Load class ‘#1’ with the current option list.

```
715 \def\LoadClassWithOptions{%
716 \@loadwithoptions\@clsextension\LoadClass}
717 \onlypreamble\LoadClassWithOptions
```

(End of definition for \LoadClassWithOptions.)

\RequirePackageWithOptions Load package ‘#1’ with the current option list.

```

718 {/2ekernel}
719 {*2ekernel | latexrelease}
720 {latexrelease}\IncludeInRelease{2020/10/01}%
721 {latexrelease} {\RequirePackageWithOptions}{Unused options issue}%
722 \def\RequirePackageWithOptions{%

```

The resetting of the unprocessed options is now done on a per package basis.

```

723 \AtEndOfPackage{\expandafter\let
724 \csname unprocessedoptions-\@currname.\@currext\endcsname
725 \relax}%
726 \@loadwithoptions\@pkgextension\RequirePackage}%
727 \onlypreamble\RequirePackageWithOptions
728 {/2ekernel | latexrelease}
729 {latexrelease}\EndIncludeInRelease

730 {latexrelease}\IncludeInRelease{0000/00/00}%
731 {latexrelease} {\RequirePackageWithOptions}{Unused options issue}%
732 {latexrelease}
733 {latexrelease}\def\RequirePackageWithOptions{%
734 {latexrelease} \AtEndOfPackage{\let\unprocessedoptions\relax}%
735 {latexrelease} \@loadwithoptions\@pkgextension\RequirePackage}%
736 {latexrelease}\EndIncludeInRelease
737 {*2ekernel}

```

(*End of definition for \RequirePackageWithOptions.*)

\usepackage To begin with, \usepackage produces an error. This is reset by \documentclass.

```

738 \def\usepackage#1{%
739 \@latex@error
740 {\noexpand \usepackage before \string\documentclass}%
741 {\noexpand \usepackage may only appear in the document
742 preamble, i.e.,\MessageBreak
743 between \noexpand\documentclass and
744 \string\begin{document}.}%
745 \gobble}
746 \onlypreamble\usepackage

```

(*End of definition for \usepackage.*)

\NeedsTeXFormat Check that the document is running on the correct system.

```

747 \def\NeedsTeXFormat#1{%
748 \def\reserved@a{#1}%
749 \ifx\reserved@a\fmtname
750 \expandafter\@needsformat
751 \else
752 \@latex@error{This file needs format ‘\reserved@a’}%
753 \MessageBreak but this is ‘\fmtname’}%
754 The current input file will not be processed
755 further,\MessageBreak
756 because it was written for some other flavor of
757 TeX.\MessageBreak\@ehd}%

```

If the file is not meant to be processed by L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  we stop inputting it, but we do not end the run. We just end inputting the current file.

```

758 \endinput \fi}
759 \onlypreamble\NeedsTeXFormat
760 \def\@needsformat{%
761 \ifnextchar[%
762 \@needsf@rmat
763 {}}
764 \onlypreamble\@needsformat

765 \def\@needsf@rmat[#1]{%
766 \ifl@t\fmtversion{#1}{}%
767 {\@latex@warning@no@line
768 {You have requested release '#1' of LaTeX,\MessageBreak
769 but only release '\fmtversion' is available}}}
770 \onlypreamble\@needsf@rmat

```

(End of definition for \NeedsTeXFormat.)

\zap@space \zap@space foo<space>\empty removes all spaces from `foo` that are not protected by { } groups.

```

771 \def\zap@space#1 #2{%
772 #1%
773 \ifx#2\empty\else\expandafter\zap@space\fi
774 #2}

```

(End of definition for \zap@space.)

\@fileswithoptions The common part of \documentclass and \usepackage.

```

775 </2ekernel>
776 <|latexrelease|\IncludeInRelease{2024/06/01}%
777 <|latexrelease| {\@fileswithoptions}{Check Group}%
778 (*2ekernel | latexrelease)
779 \def\@fileswithoptions#1{%
780 \ifnum\currentgrouplevel>z@%
781 \@latex@error
782 {Loading a class or package in a group}%
783 {Classes and packages should only be loaded at the top level}%
784 \fi
785 \ifnextchar[%]
786 {\@fileswithoptions#1}%
787 {\@fileswithoptions#1[]}}
788 </2ekernel | latexrelease>
789 <|latexrelease|\EndIncludeInRelease
790 <|latexrelease|\IncludeInRelease{0000/00/00}%
791 <|latexrelease| {\@fileswithoptions}{Check Group}%
792 <|latexrelease|\def\@fileswithoptions#1{%
793 <|latexrelease| \ifnextchar[%]
794 {\@fileswithoptions#1}%
795 {\@fileswithoptions#1[]}}
796 <|latexrelease|\EndIncludeInRelease
797 (*2ekernel)
798 \onlypreamble\@fileswithoptions

```

```

799 \def\@fileswith@ptions#1[#2]#3{%
800 \@ifnextchar[%]
801 {\@fileswith@pti@ns#1[#2]#3}%
802 {\@fileswith@pti@ns#1[#2]#3[]}}
803 \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.

```

804 </2ekernel>
805 <latexrelease>\IncludeInRelease{2020/10/01}%
806 <latexrelease> {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
807 <*2ekernel | latexrelease>
808 \def\@fileswith@pti@ns#1[#2]#3[#4]{%
809 \ifx#1\@clsextension
810 \ifx\@classoptionslist\relax
811 \protected@xdef\@classoptionslist{\zap@space#2 \empty}%

```

Save raw class list.

```

812 \gdef\@raw@classoptionslist{#2}%
813 \def\reserved@a{%
814 \onefilewithoptions#3[#2][#4]#1%
815 \documentclasshook}%
816 \else
817 \def\reserved@a{%
818 \onefilewithoptions#3[#2][#4]#1%}
819 \fi
820 \else

```

build up a list of calls to \onefilewithoptions (one for each package) without thrashing the parameter stack.

```
821 \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)

```
822 \ifx@\nnil##1\relax\else
```

If \ifx@\nnil##1@\nnil is true then #1 is (presumably) empty (Older code used \relax which is slightly easier to get into #1 by mistake, which would spoil this test.)

```

823 \ifx@\nnil##1@\nnil\else
824 \noexpand\onefilewithoptions##1[\unexpanded{#2}][#4]%
825 \noexpand\pkgextension
826 \fi
827 \expandafter\reserved@b
828 \fi}%
829 \edef\reserved@a{\zap@space#3 \empty}%
830 \edef\reserved@a{\expandafter\reserved@b\reserved@a,\nnil,}%

```

```

831 \fi
832 \reserved@a}
833 </2ekernel | latexrelease>
834 <latexrelease>\EndIncludeInRelease
835 <latexrelease>\IncludeInRelease{2017/01/01}%
836 <latexrelease> {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
837 <latexrelease>\def\@fileswith@pti@ns#1[#2]#3[#4]{%
838 <latexrelease> \ifx#1\@clsextension
839 <latexrelease> \ifx\@classoptionslist\relax
840 <latexrelease> \xdef\@classoptionslist{\zap@space#2 \empty}%
841 <latexrelease> \def\reserved@a{%
842 <latexrelease> \@onelinewithoptions#3[#2] [#4]#1%
843 <latexrelease> \@documentclasshook}%
844 <latexrelease> \else
845 <latexrelease> \def\reserved@a{%
846 <latexrelease> \@onelinewithoptions#3[#2] [#4]#1}%
847 <latexrelease> \fi
848 <latexrelease> \else
849 <latexrelease> \def\reserved@b##1{%
850 <latexrelease> \ifx\@nnil##1\relax\else
851 <latexrelease> \ifx\@nnil##1\@nnil\else
852 <latexrelease> \noexpand\@onelinewithoptions##1[#2] [#4]%
853 <latexrelease> \noexpand\@pkgextension
854 <latexrelease> \fi
855 <latexrelease> \expandafter\reserved@b
856 <latexrelease> \fi}%
857 <latexrelease> \edef\reserved@a{\zap@space#3 \empty}%
858 <latexrelease> \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nnil,}%
859 <latexrelease> \fi
860 <latexrelease> \reserved@a}
861 <latexrelease>\EndIncludeInRelease
862 <latexrelease>\IncludeInRelease{0000/00/00}%
863 <latexrelease> {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
864 <latexrelease>\def\@fileswith@pti@ns#1[#2]#3[#4]{%
865 <latexrelease> \ifx#1\@clsextension
866 <latexrelease> \ifx\@classoptionslist\relax
867 <latexrelease> \xdef\@classoptionslist{\zap@space#2 \empty}%
868 <latexrelease> \def\reserved@a{%
869 <latexrelease> \@onelinewithoptions#3[#2] [#4]#1%
870 <latexrelease> \@documentclasshook}%
871 <latexrelease> \else
872 <latexrelease> \def\reserved@a{%
873 <latexrelease> \@onelinewithoptions#3[#2] [#4]#1}%
874 <latexrelease> \fi
875 <latexrelease> \else
876 <latexrelease> \def\reserved@b##1{%
877 <latexrelease> \ifx\@nil##1\relax\else
878 <latexrelease> \ifx\relax##1\relax\else
879 <latexrelease> \noexpand\@onelinewithoptions##1[#2] [#4]%
880 <latexrelease> \noexpand\@pkgextension
881 <latexrelease> \fi
882 <latexrelease> \expandafter\reserved@b
883 <latexrelease> \fi}%
884 <latexrelease> \edef\reserved@a{\zap@space#3 \empty}%

```

```

885 〈\latexrelease〉 \edef\reserved@a{%
886 〈\latexrelease〉 \expandafter\reserved@b\reserved@a,\@nil,}%
887 〈\latexrelease〉 \fi
888 〈\latexrelease〉 \reserved@a}
889 〈\latexrelease〉\EndIncludeInRelease
890 〈*2ekernel〉
891 〈@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.

892 〈/2ekernel〉
893 〈*2ekernel | \latexrelease〉
894 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
895 〈\latexrelease〉 {〈\@onefilewithoptions〉{Hooks and unused options issue}}%

```

Here this macro is called `\@onefilewithoptions`, but further ahead in this file it is renamed to `\load@onefilewithoptions`, and `\@onefilewithoptions` becomes a wrapper around this, used for bookkeeping when rolling back. Therefore, when in `\latexrelease`, we need to define `\load@onefilewithoptions` instead, thus the extra guarded `\def` line below:

```

896 〈*2ekernel〉
897 〈\def\@onefilewithoptions#1[#2] [#3]#4{%
898 〈/2ekernel〉
899 〈\latexrelease〉\def\load@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 substitution 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.

```

900 〈\@expl@@@filehook@set@curr@file@nNN{#1.#4}\reserved@a\reserved@b
901 〈\edef\reserved@c{\def\noexpand\reserved@c####1}%
902 〈\detokenize\expandafter{\expanded{. #4}}%
903 〈\noexpand\@nil{\def\noexpand\reserved@a{\####1}}\reserved@c
904 〈\expandafter\reserved@c\reserved@a\@nil
905 〈\@pushfilename
906 〈\xdef\@currname{\string\@makeletter\reserved@a}%
907 〈\xdef\@currpath{\ifx\reserved@b\@empty\else\reserved@b\fi}%
908 〈\global\let\@currext\@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.

```
909 \@ifl@aded\@currname
```

In the current preferred approach, a key family name will exist for processing using `\tkeys`. In that case, we replace the previous package options with the new ones, then call the key handler. Otherwise, we use the more classical clash handler.

```
910 {%
911 \@ifundefined{opt@handler@\@currname.\@currname}%
912 {\@onefilewithoptions@clashchk{\#2}}%
913 {%
914 \expandafter\protected@edef
915 \csname opt@\@currname.\@currname\endcsname
916 {\zap@space\#2 \empty}%
917 \expandafter\def
918 \csname @raw@opt@\@currname.\@currname\expandafter\endcsname
919 \expandafter{\#2}%
920 \nameuse{opt@handler@\@currname.\@currname}%
921 }%
922 }%
923 {\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!

```
924 \@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`.

```
925 \IfFileExists{\@currpath\@currname.\@currname}{}{%
926 \@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.

```
927 \ifx\@currname\empty
928 \expandafter\@gobble
929 \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.

```
930 \@disable@packageload@do{\@currname.\@currname}%
931 {\@expl@@@filehook@file@push@@
932 \set@curr@file{\@currpath\@currname.\@currname}%
933 \@filehook@set@CurrentFile}
```

The `\set@curr@file` line above might have replaced the file, so `\@currname` and `\@currname` 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.

```

934 \expandafter\@swaptwoargs\expandafter
935 {\expandafter{\@currpkg@reqd}}%
936 {%
937 \edef\@currpkg@reqd{\@currname.\@currext}%
938 \ifx\CurrentFile\CurrentFileUsed
939 \else
940 \filename@parse\@curr@file
941 \edef\@currpath{\string@makeletter\filename@area}%
942 \edef\@currname{\string@makeletter\filename@base}%
943 \edef\@currext{\string@makeletter\filename@ext}%
944 \fi
945 \load@onefile@withoptions{#2}%
946 \def\@currpkg@reqd{\@currpkg@reqd}%
947 }%

```

Now just clean up and exit.

```

948 \expl@@@filehook@file@pop@@}%
949 \expandafter\@firstofone
950 \fi}%

```

Except in the case where `\@currname` is empty, the date is checked against the date marked in the package file:

```

951 {\@ifl@ter\@currext{\@currname}{#3}{}}%
952 {\@latex@warning@no@line
953 {You have requested,\on@line,
954 version\MessageBreak
955 '#3' of \cls@pkg\space \@currname,\MessageBreak
956 but only version\MessageBreak
957 '\csname ver@\@currname.\@currext\endcsname'\MessageBreak
958 is available}}%
959
960 \ifx\@currext\@clsextension\let\LoadClass\@twoloadclasserror\fi}%
961 \popfilename
962 \reset@options

```

If the package is already loaded, check that there were no option clashes.

```

962 \def\@onefilewithoptions@clashchk#1{%
963 \@if@ptions\@currext{\@currname}{#1}{}}%
964 {\@latex@error
965 {Option clash for \cls@pkg\space \@currname}%
966 {The package \@currname\space has already been loaded
967 with options:\MessageBreak
968 \space\space[\optionlist{\@currname.\@currext}]\MessageBreak
969 There has now been an attempt to load it
970 with options\MessageBreak
971 \space\space[#1]\MessageBreak
972 Adding the global options:\MessageBreak
973 \space\space

```

```

974 \optionlist{\@currname.\@currext},#1\MessageBreak
975 to your \noexpand\documentclass declaration may fix this.%

976 \MessageBreak
977 Try typing \space <return> \space to proceed.}%
978 \@firstofone}

979 \let\@currpkg@reqd\@empty
980 \onlypreamble\onefilewithoptions
 The kernel no longer uses \unprocessedoptions
981 \let\@unprocessedoptions\@undefined

```

Now the action taken when a file is not found. Path must be included here as it eventually leads to a file lookup.

```

982 \def\@missing@onefilewithoptions#1{%
983 \@missingfileerror{\@currpath\@currname}\@currext
984 \global\let\@currpath\@missingfile@area
985 \global\let\@currname\@missingfile@base
986 \global\let\@currext\@missingfile@ext}

```

Now the code that actually does the file loading:

```

\load\onefile@withoptions 987 \def\load@onefile@withoptions#1{%
988 \let\CurrentOption\@empty
989 \reset@options

```

Grab everything in a macro, so the parameter stack is popped before any processing begins.

```

990 \def\reserved@a{%
991 \pass@options\@currext{#1}{\@currname}%
992 \expandafter\let
993 \csname opt@\@currpkg@reqd\expandafter\endcsname
994 \csname opt@\@currname.\@currext\endcsname
995 \expandafter\let
996 \csname @raw@opt@\@currpkg@reqd\expandafter\endcsname
997 \csname @raw@opt@\@currname.\@currext\endcsname
998 \global\expandafter
999 \let\csname ver0@\@currname.\@currext\endcsname\@empty

```

We initialize \....-h@k here and only if we load the file so that it remains undefined otherwise.

```
1000 \expandafter\let\csname\@currname.\@currext-h@k\endcsname\@empty
```

When the current extension is \@pkgeextension 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.

```

1001 %-----

1002 \ifx\@currext\@pkgeextension
1003 \UseHook{package/before}%
1004 \UseOneTimeHook{package/\@currname/before}%
1005 \else
1006 \ifx\@currext\@clsextension
1007 \UseHook{class/before}%
1008 \UseOneTimeHook{class/\@currname/before}%
1009 \fi
1010 \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.

```
1011 \InputIfFileExists{\@currpath\@currpkg@reqd}{}%
1012 {\@latex@error
1013 {The \@cls@pkg\space\@currpkg@reqd\space failed to load}\@ehd}%
1014 %-----
```

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 interference in case of nested packages. The whole code for handling this issue (GitHub 22) was provided by Hironobu Yamashita, thanks for that.

```
1015 \expandafter\let\csname unprocessedoptions-\@currname.\@currext\endcsname
1016 \@@unprocessedoptions
1017 \csname\@currname.\@currext-h@k\endcsname
1018 \expandafter\let\csname\@currname.\@currext-h@k\endcsname
1019 \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.

```
1020 \ifx\@unprocessedoptions\relax
1021 \let\@unprocessedoptions\undefined
```

Otherwise run the per package set of unused options.

```
1022 \else
1023 \csname unprocessedoptions-\@currname.\@currext\endcsname
1024 \fi
```

In either case we drop the macro afterwards as it is no longer needed.

```
1025 \expandafter\let
1026 \csname unprocessedoptions-\@currname.\@currext\endcsname
1027 \undefined
```

And same procedure, James, when we are finished loading, except that the hook order is now reversed.

```
1028 %-----
1029 \ifx\@currext\@pkgextension
1030 \UseOneTimeHook{package/\@currname/after}%
1031 \UseHook{package/after}%
1032 \else
1033 \ifx\@currext\@clsextension
1034 \UseOneTimeHook{class/\@currname/after}%
1035 \UseHook{class/after}%
1036 \fi
1037 \fi}%
1038 %-----
1039 \@ifl@aded\@currext\@currname{}{\reserved@a}
```

Now declare the non-generic package and class hooks used above:

```
1040 \NewHook{package/before}
1041 \NewHook{class/before}
1042 \NewReversedHook{package/after}
1043 \NewReversedHook{class/after}
```

```

1044 {/2ekernel | latexrelease}
1045 \end{IncludeInRelease}
1046 \IncludeInRelease{0000/00/00}%
1047 {\@onefilewithoptions}{Hooks and unused options issue}%
1048 \end{IncludeInRelease}

```

Because of the way `\@onefilewithoptions` is changed for rollback handling below we have to define `\load@onefilewithoptions` when rolling back!

```

1049 \def\load@onefilewithoptions#1[#2][#3]{%
1050 \pushfilename
1051 \xdef\currname{#1}%
1052 \global\let\currext\#4%
1053 \let\CurrentOption\empty
1054 \resetoptions
1055 \makeatletter
1056 \def\reserved@a{%
1057 \@if@aded{\currext{#1}%
1058 {\@if@ptions{\currext{#1}{#2}{}}{%
1059 {\@latex@error
1060 {Option clash for \cls@pkg\space #1}%
1061 {The package #1 has already been loaded
1062 with options:\MessageBreak
1063 \space\space[\optionlist{#1.\currext}]\MessageBreak
1064 There has now been an attempt to load it
1065 with options\MessageBreak
1066 \space\space[#2]\MessageBreak
1067 Adding the global options:\MessageBreak
1068 \space\space
1069 \optionlist{#1.\currext},#2\MessageBreak
1070 to your \noexpand\documentclass declaration may fix this.%\MessageBreak
1071 Try typing \space <return> \space to proceed.}}}}%
1072 {\@pass@ptions{\currext{#2}{#1}}%
1073 \global\expandafter
1074 \let\csname ver@\currname.\currext\endcsname\empty
1075 \expandafter\let\csname\currname.\currext-h@k\endcsname\empty
1076 \InputIfFileExists
1077 {\currname.\currext}%
1078 {}%
1079 {\@missingfileerror\currname\currext}%
1080 \let\unprocessedoptions\@unprocessedoptions
1081 \csname\currname.\currext-h@k\endcsname
1082 \expandafter\let\csname\currname.\currext-h@k\endcsname
1083 \undefined
1084 \@unprocessedoptions\%%
1085 \@if@ter{\currext{#1}{#3}{}}%
1086 {\@latex@warning@no@line
1087 {You have requested,\on@line,
1088 version\MessageBreak
1089 '#3' of \cls@pkg\space #1,\MessageBreak
1090 but only version\MessageBreak
1091 '\csname ver@\#1.\currext\endcsname'\MessageBreak
1092 is available}}%
1093 \ifx\currext\clsextension\let\LoadClass\@twoloadclasserror\fi
1094

```

```

1095 〈latexrelease〉 \popfilename
1096 〈latexrelease〉 \reset@ptions}%
1097 〈latexrelease〉 \reserved@a}
1098 〈latexrelease〉
1099 〈latexrelease〉\let \load@oneline@withoptions \undefined
1100 〈latexrelease〉\let \missing@oneline@withoptions \undefined
1101 〈latexrelease〉
1102 〈latexrelease〉\EndIncludeInRelease
1103 {*2ekernel}

```

(End of definition for `\@files@with@options` and others.)

`\@@files@with@pti@ns` Save the definition (for error checking).

```

1104 \let\@@files@with@pti@ns\@files@with@pti@ns
1105 \onlypreamble\@@files@with@pti@ns

```

(End of definition for `\@@files@with@pti@ns`.)

`\@reset@ptions` Reset the default option, and clear lists of declared options.

```

1106 \def\@reset@ptions{%
1107 \global\ifx\@currext\@clsextension
1108 \let\default@ds\OptionNotUsed
1109 \else
1110 \let\default@ds\@unknownoptionerror
1111 \fi
1112 \global\let\ds@\emptyset
1113 \global\let\@declaredoptions\emptyset}
1114 \onlypreamble\@reset@ptions

```

(End of definition for `\@reset@ptions`.)

## 4.1 Hooks

Allow code to be saved to be executed at specific later times.

Here we save things in macros. I considered using toks registers (and `\addto@hook` from the NFSS code), but that would require stacking the contents in the case of required packages, so just generate a new macro for each package.

`\@begindocumenthook` Stuff to appear at the beginning or end of the document.

```

1115 \ifx\@begindocumenthook\undefined
1116 \let\@begindocumenthook\emptyset
1117 \fi
1118 \let\@enddocumenthook\emptyset

```

(End of definition for `\@begindocumenthook` and `\@enddocumenthook`.)

`\AtEndOfPackage` The access functions.

```

1119 \def\AtEndOfPackage{%
1120 \expandafter\g@addto@macro\csname\@currname.\@currext-h@ok\endcsname}%
1121 \let\AtEndOfClass\AtEndOfPackage
1122 \onlypreamble\AtEndOfPackage
1123 \onlypreamble\AtEndOfClass

```

```

1124 </2ekernel>
1125 <*2ekernel | latexrelease>
1126 <latexrelease>\IncludeInRelease{2020/10/01}%
1127 <latexrelease> {\AtBeginDocument}{Use hook system}%
1128 \DeclareRobustCommand\AtBeginDocument{\AddToHook{begindocument}}
1129 \DeclareRobustCommand\AtEndDocument {\AddToHook{enddocument}}%
1130 \% \DeclareRobustCommand\AtEndDocument {\AddToHook{env/document/end}} % alternative impl
1131 </2ekernel | latexrelease>
1132 <latexrelease>\EndIncludeInRelease
1133 <latexrelease>\IncludeInRelease{0000/00/00}%
1134 <latexrelease> {\AtBeginDocument}{Use hook system}%
1135 <latexrelease>
1136 <latexrelease>\DeclareRobustCommand\AtBeginDocument{\g@addto@macro\@begindocumenthook}
1137 <latexrelease>\DeclareRobustCommand\AtEndDocument{\g@addto@macro\@enddocumenthook}
1138 <latexrelease>
1139 <latexrelease>\EndIncludeInRelease
1140 <*2ekernel>
1141 \onlypreamble\AtBeginDocument

```

(*End of definition for \AtEndOfPackage and others.*)

\@cls@pkg The current file type.

```

1142 </2ekernel>
1143 <*2ekernel | latexrelease>
1144 <latexrelease>\IncludeInRelease{2024/11/01}%
1145 <latexrelease> {\@cls@pkg}{Allow for more extensions}%
1146 \def\@cls@pkg{%
1147 \ifx\@currext\@clsextension
1148 document class%
1149 \else
1150 \ifx\@currext\@pkgextension
1151 package%
1152 \else
1153 file%
1154 \fi
1155 \fi}
1156 </2ekernel | latexrelease>
1157 <latexrelease>\EndIncludeInRelease
1158 <latexrelease>\IncludeInRelease{0000/00/00}%
1159 <latexrelease> {\@cls@pkg}{Allow for more extensions}%
1160 <latexrelease>
1161 <latexrelease>\def\@cls@pkg{%
1162 <latexrelease> \ifx\@currext\@clsextension
1163 document class%
1164 <latexrelease> \else
1165 package%
1166 <latexrelease> \fi}
1167 <latexrelease>\EndIncludeInRelease
1168 <*2ekernel>
1169 \onlypreamble\@cls@pkg

```

(*End of definition for \@cls@pkg.*)

\@unknownoptionerror Bad option.

```

1170 \def\@unknownoptionerror{%
1171 \@latex@error
1172 {Unknown option ‘\CurrentOption’ for \cls@pkg\space‘\currname’}%
1173 {The option ‘\CurrentOption’ was not declared in
1174 \cls@pkg\space‘\currname’, perhaps you\MessageBreak
1175 misspelled its name.
1176 Try typing \space <return>
1177 \space to proceed.}%
1178 \onlypreamble\@unknownoptionerror

```

(End of definition for \@unknownoptionerror.)

\@@unprocessedoptions Declare an error for each option, unless a \ProcessOptions occurred.

```

1179 \def\@@unprocessedoptions{%
1180 \ifx\@currext\@pkextension
1181 \protected@edef\@curroptions{\optionlist{\currname.\@currext}}%
1182 \@for\CurrentOption:=\curroptions\do{%
1183 \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
1184 \fi}
1185 \onlypreamble\@unprocessedoptions
1186 \onlypreamble\@@unprocessedoptions

```

(End of definition for @@unprocessedoptions.)

\@badrequireerror \RequirePackage or \LoadClass occurs in the options section.

```

1187 \def\@badrequireerror#1[#2]#3[#4]{%
1188 \@latex@error
1189 {\noexpand\RequirePackage or \noexpand\LoadClass
1190 in Options Section}%
1191 {The \cls@pkg\space ‘\currname’ is defective.\MessageBreak
1192 It attempts to load ‘#3’ in the options section, i.e.,\MessageBreak
1193 between \noexpand\DeclareOption and \string\ProcessOptions.}%
1194 \onlypreamble\@badrequireerror

```

(End of definition for \@badrequireerror.)

\@twoloadclasserror Two \LoadClass in a class.

```

1195 \def\@twoloadclasserror{%
1196 \@latex@error
1197 {Two \noexpand\LoadClass commands}%
1198 {You may only use one \noexpand\LoadClass in a class file}%
1199 \onlypreamble\@twoloadclasserror

```

(End of definition for \@twoloadclasserror.)

\@twoclasseserror Two \documentclass or \documentstyle.

```

1200 \def\@twoclasseserror#1{%
1201 \@latex@error
1202 {Two \noexpand\documentclass or \noexpand\documentstyle commands}%
1203 {The document may only declare one class.}\@gobble}
1204 \onlypreamble\@twoclasseserror

```

(End of definition for \@twoclasseserror.)

## 4.2 Providing shipment

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

```
1205 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}
```

(End of definition for \two@digits.)

`file\filecontents`) This environment implements inline files. The star-form does not write extra comments  
`\endfilecontents` into the file.

```
1206 {/2ekernel}
1207 {*2ekernel | latexrelease}
1208 {latexrelease}\IncludeInRelease{2020/10/01}%
1209 {latexrelease} {\filec@ntents}{Define \q@curr@file directly (gh/220)}%
1210 %
```

We use `@tempswa` to mean no preamble writing and reuse `@files w` to indicate no overwriting:

```
1211 \def\filecontents{\@tempswatrue\@filestrue
1212 \@ifnextchar[\filec@ntents@opt\filec@ntents
1213 }
1214 \@namedef{filecontents*}{\@tempswafalse\@filestrue
1215 \@ifnextchar[\filec@ntents@opt\filec@ntents
1216 }
```

To handle the optional argument we execute for each option the command `\filec@ntents@OPTION` if it exist or complain about unknown option.

```
1217 \def\filec@ntents@opt[#1]{%
1218 \edef\@fortmp{\zap@space#1 \empty}%
1219 \@for\reserved@a:=\@fortmp\do{%
1220 \ifcsname filec@ntents@\reserved@a\endcsname
1221 \csname filec@ntents@\reserved@a\endcsname
1222 \else
1223 \@latex@error{Unknown filecontents option \reserved@a}%
1224 {Valid options are force (or overwrite), nosearch, noheader, nowarn}%
1225 \fi}%
1226 \filec@ntents
1227 }
```

Option `force` (or `overwrite`) changes the overwriting switch

```
1228 \let\filec@ntents@force\@filesfalse
1229 \let\filec@ntents@overwrite\@filesfalse % alternative name
```

and option `noheader` the preamble switch (which is equivalent to using the star form of the environment).

```
1230 \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.

```
1231 \def\filec@ntents@nosearch{%
1232 \let\filec@ntents@checkdir\@currdir
1233 \def\filec@ntents@where{in current directory}}
```

By default we search the whole tree:

```
1234 \let\filec@ntents@checkdir\empty
1235 \def\filec@ntents@where{exists on the system}
```

Option `nowarn` does not show any warning on the terminal but still writes it to the `.log`.

```
1236 \def\filec@ntents@nowarn{%
1237 \let\filec@ntents@warning\@latex@note@no@line
1238 }
```

By default we show terminal warnings.

```
1239 \let\filec@ntents@warning\@latex@warning@no@line
1240 \begingroup%
1241 \tempcnta=1
1242 \loop
1243 \catcode\tempcnta=12 %
1244 \advance\tempcnta\one %
1245 \ifnum\tempcnta<32 %
1246 \repeat %
1247 \catcode`*=11 %
1248 \catcode`\^M\active%
1249 \catcode`\^L\active\let^L\relax%
1250 \catcode`\^I\active%

1251 \gdef\filec@ntents#1{%
1252 \set@curr@file{\filec@ntents@checkdir#1}%
1253 \edef\q@curr@file{"\curr@file"}%
```

LuaTeX has more writes (and 18 is safe here).

```
1254 \chardef\reserved@c\ifx\directlua\undefined 15 \else 127 \fi%
1255 \openin@\inputcheck\q@curr@file \space %
1256 \ifeof\inputcheck%
1257 \@latex@note@no@line%
1258 {Writing file '\currdir\curr@file'}%
1259 \ch@ck7\reserved@c\write\relax%
1260 \immediate\openout\reserved@c\q@curr@file\relax%
1261 \else%
1262 \if@files%
1263 \@latex@note@no@line%
1264 {File '\curr@file' already \filec@ntents@where.\MessageBreak%
1265 Not generating it from this source}%
1266 \let\write@gobbletwo%
1267 \let\closeout@gobble%
1268 \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!

```
1269 \edef\reserved@b{\detokenize\expandafter{\jobname}}%
1270 \ifx\curr@file\reserved@b%
1271 \if@files%
1272 \else%
1273 \edef\reserved@b{\reserved@b\detokenize{.tex}}%
1274 \ifx\curr@file\reserved@b
1275 \if@files%
1276 \fi%
1277 \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.

```

1278 \ch@ck7\reserved@c\write\relax%
1279 \if@filesw% % Foul ... trying to overwrite \jobname!
1280 \@latex@error{Trying to overwrite '\jobname.tex'}{You can't %
1281 write to the file you are reading from!\MessageBreak%
1282 Data is written to screen instead.}%
1283 \else%
1284 \filec@ntents@warning%
1285 {[Writing or overwriting file '@curr@file']}%
1286 \immediate\openout\reserved@c\q@curr@file\relax%
1287 \fi%
1288 \fi%
1289 \fi%

```

Closing the \@inputcheck is done here to avoid having to do this in each branch.

```

1290 \closein\@inputcheck%
1291 \if@tempswa%
1292 \immediate\write\reserved@cf%
1293 \@percentchar\@percentchar\space%
1294 \expandafter\@gobble\string\LaTeX2e file '@curr@file'^^J%
1295 \@percentchar\@percentchar\space generated by the %
1296 '@currenvir' \expandafter\@gobblefour\string\newenvironment^^J%
1297 \@percentchar\@percentchar\space from source '\jobname' on %
1298 'number\year/\two@digits\month/\two@digits\day.^~J%
1299 \@percentchar\@percentchar}%
1300 \fi%
1301 \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.

```

1302 \count@ 128\relax%
1303 \loop%
1304 \catcode\count@ 11\relax%
1305 \advance\count@ \@ne%
1306 \ifnum\count@<\@cclvi%
1307 \repeat%
1308 \edef\E{\@backslashchar end\string{@currenvir\string}}%
1309 \edef\reserved@b{%
1310 \def\noexpand\reserved@b{%
1311 #####1\@#####2\@#####3\relax}%
1312 \reserved@b{%
1313 \ifx\relax##3\relax%

```

There was no `\end{filecontents}`

```

1314 \immediate\write\reserved@c{##1}%
1315 \else%

```

There was a `\end{filecontents}`, so stop this time.

```

1316 \edef^M{\noexpand\end{@currenvir}}%
1317 \ifx\relax##1\relax%
1318 \else%

```

Text before the \end, write it with a warning.

```
1319 \Q@latex@warning{Writing text ‘##1’ before %
1320 \string\end{\@currenvir}\MessageBreak
1321 as last line of \curr@file}%
1322 \immediate\write\reserved@c{##1}%
1323 \fi%
1324 \ifx\relax##2\relax%
1325 \else%
```

Text after the \end, ignore it with a warning.

```
1326 \Q@latex@warning{%
1327 Ignoring text ‘##2’ after \string\end{\@currenvir}}%
1328 \fi%
1329 \fi%
1330 ^~M}%
1331 \catcode`^~L\active%
1332 \let\L\@undefined%
1333 \def^~L{\expandafter\ifx\csname L\endcsname\relax\fi ^~J^~J}%
1334 \catcode`^~I\active%
1335 \let\I\@undefined%
1336 \def^~I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
1337 \catcode`^~M\active%
1338 \edef^~M##1^~M{%
1339 \noexpand\reserved@b##1\relax}%
1340 \endgroup%
1341 </2ekernel | latexrelease>
1342 <latexrelease>\EndIncludeInRelease
1343 <latexrelease>\IncludeInRelease{2019/10/01}%
1344 <latexrelease> {\filec@ntents}{Spaces in file names + optional arg}%
1345 <latexrelease>
1346 <latexrelease>\def\filecontents{\@tempswatru@filestrue
1347 <latexrelease> \@ifnextchar[\filec@ntents@opt\filec@ntents
1348 <latexrelease>]
1349 <latexrelease>\Qnamedef{filecontents*}{\@tempswafalse\@filestrue
1350 <latexrelease> \@ifnextchar[\filec@ntents@opt\filec@ntents
1351 <latexrelease>]
1352 <latexrelease>\def\filec@ntents@opt[#1]{%
1353 <latexrelease> \edef\@fortmp{\zap@space#1 \empty}%
1354 <latexrelease> \@for\reserved@a:=\@fortmp\do{%
1355 <latexrelease> \ifcsname filec@ntents@\reserved@a\endcsname
1356 <latexrelease> \csname filec@ntents@\reserved@a\endcsname
1357 <latexrelease> \else
1358 <latexrelease> \Q@latex@error{Unknown filecontents option \reserved@a}%
1359 <latexrelease> {Valid options are force (or overwrite), nosearch, noheader}%
1360 <latexrelease> \fi}%
1361 <latexrelease> \filec@ntents
1362 <latexrelease>}%
1363 <latexrelease>\let\filec@ntents@force\@filesfalse
1364 <latexrelease>\let\filec@ntents@overwrite\@filesfalse % alternative name
1365 <latexrelease>\let\filec@ntents@noheader\@tempswafalse
1366 <latexrelease>\def\filec@ntents@nosearch{%
1367 <latexrelease> \let\filec@ntents@checkdir\@currdir
1368 <latexrelease> \def\filec@ntents@where{in current directory}}
```

```

1369 ⟨latexrelease⟩\let\filec@ntents@checkdir@empty
1370 ⟨latexrelease⟩\def\filec@ntents@where{exists on the system}
1371 ⟨latexrelease⟩\begingroup%
1372 ⟨latexrelease⟩\@tempcnta=1
1373 ⟨latexrelease⟩\loop
1374 ⟨latexrelease⟩ \catcode\@tempcnta=12 %
1375 ⟨latexrelease⟩ \advance\@tempcnta\@ne %
1376 ⟨latexrelease⟩\ifnum\@tempcnta<32 %
1377 ⟨latexrelease⟩\repeat %
1378 ⟨latexrelease⟩\catcode`*=11 %
1379 ⟨latexrelease⟩\catcode`\^M\active%
1380 ⟨latexrelease⟩\catcode`\^L\active\let\^L\relax%
1381 ⟨latexrelease⟩\catcode`\^I\active%
1382 ⟨latexrelease⟩\gdef\filec@ntents#1{%
1383 ⟨latexrelease⟩ \set@curr@file{\filec@ntents@checkdir#1}%
1384 ⟨latexrelease⟩ \edef\q@curr@file{\expandafter\quote@name\expandafter{\@curr@file}}%
1385 ⟨latexrelease⟩ \chardef\reserved@c\ifx\directlua\undefined 15 \else 127 \fi%
1386 ⟨latexrelease⟩ \openin\@inputcheck\q@curr@file \space %
1387 ⟨latexrelease⟩ \ifeof\@inputcheck%
1388 ⟨latexrelease⟩ \ @latex@warning@no@line%
1389 ⟨latexrelease⟩ {Writing file '\@currdir\@curr@file'}%
1390 ⟨latexrelease⟩ \ch@ck7\reserved@c\write\relax%
1391 ⟨latexrelease⟩ \immediate\openout\reserved@c\q@curr@file\relax%
1392 ⟨latexrelease⟩ \else%
1393 ⟨latexrelease⟩ \if@files w%
1394 ⟨latexrelease⟩ \ @latex@warning@no@line%
1395 ⟨latexrelease⟩ {File '\@curr@file' already \filec@ntents@where.\MessageBreak%
1396 ⟨latexrelease⟩ Not generating it from this source}%
1397 ⟨latexrelease⟩ \let\write@\gobbletwo%
1398 ⟨latexrelease⟩ \let\closeout\gobble%
1399 ⟨latexrelease⟩ \else%
1400 ⟨latexrelease⟩ \edef\reserved@a{#1}%
1401 ⟨latexrelease⟩ \edef\reserved@a{\detokenize\expandafter{\reserved@a}}%
1402 ⟨latexrelease⟩ \edef\reserved@b{\detokenize\expandafter{\jobname}}%
1403 ⟨latexrelease⟩ \ifx\reserved@a\reserved@b%
1404 ⟨latexrelease⟩ \ @files wtrue%
1405 ⟨latexrelease⟩ \else%
1406 ⟨latexrelease⟩ \edef\reserved@b{\reserved@b\detokenize{.tex}}%
1407 ⟨latexrelease⟩ \ifx\reserved@a\reserved@b%
1408 ⟨latexrelease⟩ \ @files wtrue%
1409 ⟨latexrelease⟩ \fi%
1410 ⟨latexrelease⟩ \fi%
1411 ⟨latexrelease⟩ \ch@ck7\reserved@c\write\relax%
1412 ⟨latexrelease⟩ \if@files w% Foul ... trying to overwrite \jobname!
1413 ⟨latexrelease⟩ \Q@late@error{Trying to overwrite '\jobname.tex'}{You can't %
1414 ⟨latexrelease⟩ write to the file you are reading from!\MessageBreak%
1415 ⟨latexrelease⟩ Data is written to screen instead.}%
1416 ⟨latexrelease⟩ \else%
1417 ⟨latexrelease⟩ \ @latex@warning@no@line%
1418 ⟨latexrelease⟩ {Writing or overwriting file '\@currdir\@curr@file'}%
1419 ⟨latexrelease⟩ \immediate\openout\reserved@c\q@curr@file\relax%
1420 ⟨latexrelease⟩ \fi%
1421 ⟨latexrelease⟩ \fi%
1422 ⟨latexrelease⟩ \fi%

```

```

1423 <{latexrelease}> \closein\@inputcheck%
1424 <{latexrelease}> \if@tempswa%
1425 <{latexrelease}> \immediate\write\reserved@c{%
1426 \@percentchar\@percentchar\space%
1427 \expandafter\@gobble\string\LaTeX2e file '\@curr@file'^^J%
1428 \@percentchar\@percentchar\space generated by the %
1429 '@currenvir' \expandafter\@gobblefour\string\newenvironment^^J%
1430 \@percentchar\@percentchar\space from source '\jobname' on %
1431 \number\year/\two@digits\month/\two@digits\day.^^J%
1432 \@percentchar\@percentchar}%
1433 <{latexrelease}> \fi%
1434 <{latexrelease}> \let\do\@makeother\dospecials%
1435 <{latexrelease}> \count@ 128\relax%
1436 <{latexrelease}> \loop%
1437 <{latexrelease}> \catcode\count@ 11\relax%
1438 <{latexrelease}> \advance\count@ \@ne%
1439 <{latexrelease}> \ifnum\count@<\@ccvi%
1440 <{latexrelease}> \repeat%
1441 <{latexrelease}> \edef\E{\@backslashchar end\string{\@currenvir\string}}%
1442 <{latexrelease}> \edef\reserved@b{%
1443 <{latexrelease}> \def\noexpand\reserved@b{%
1444 <{latexrelease}> #####1\E#####2\E#####3\relax}%
1445 <{latexrelease}> \reserved@b{%
1446 <{latexrelease}> \ifx\relax##3\relax%
1447 <{latexrelease}> \immediate\write\reserved@c{##1}%
1448 <{latexrelease}> \else%
1449 <{latexrelease}> \edef^^M{\noexpand\end{\@currenvir}}%
1450 <{latexrelease}> \ifx\relax##1\relax%
1451 <{latexrelease}> \else%
1452 <{latexrelease}> \@latex@warning{Writing text '##1' before %
1453 <{latexrelease}> \string\end{\@currenvir}\MessageBreak as last line of \@curr@file}%
1454 <{latexrelease}> \immediate\write\reserved@c{##1}%
1455 <{latexrelease}> \fi%
1456 <{latexrelease}> \ifx\relax##2\relax%
1457 <{latexrelease}> \else%
1458 <{latexrelease}> \@latex@warning{%
1459 <{latexrelease}> Ignoring text '##2' after \string\end{\@currenvir}}%
1460 <{latexrelease}> \fi%
1461 <{latexrelease}> \fi%
1462 <{latexrelease}> ^^M}%
1463 <{latexrelease}> \catcode`^\active%
1464 <{latexrelease}> \let\@undefined%
1465 <{latexrelease}> \def`^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^J}%
1466 <{latexrelease}> \catcode`^\active%
1467 <{latexrelease}> \let\@I\@undefined%
1468 <{latexrelease}> \def`^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
1469 <{latexrelease}> \catcode`^\active%
1470 <{latexrelease}> \edef`^M#1`^M{%
1471 <{latexrelease}> \noexpand\reserved@b##1\@E\@E\relax}%
1472 <{latexrelease}> \endgroup%
1473 <{latexrelease}> \EndIncludeInRelease%
1474 <{latexrelease}> \IncludeInRelease{0000/00/00}%
1475 <{latexrelease}> {\filecontents}{Spaces in file names + optional arg}%
1476 <{latexrelease}>

```

```

1477 〈latexrelease〉\let\filec@ntents@opt \undefined
1478 〈latexrelease〉\let\filec@ntents@force \undefined
1479 〈latexrelease〉\let\filec@ntents@overwrite \undefined
1480 〈latexrelease〉\let\filec@ntents@noheader \undefined
1481 〈latexrelease〉\let\filec@ntents@nosearch \undefined
1482 〈latexrelease〉\let\filec@ntents@checkdir \undefined
1483 〈latexrelease〉\let\filec@ntents@where \undefined
1484 〈latexrelease〉
1485 〈latexrelease〉\begingroup%
1486 〈latexrelease〉\@tempcnta=1
1487 〈latexrelease〉\loop
1488 〈latexrelease〉 \catcode\@tempcnta=12 %
1489 〈latexrelease〉 \advance\@tempcnta\@ne %
1490 〈latexrelease〉\ifnum\@tempcnta<32 %
1491 〈latexrelease〉\repeat %
1492 〈latexrelease〉\catcode`*=11 %
1493 〈latexrelease〉\catcode`\^M\active%
1494 〈latexrelease〉\catcode`\^L\active\let`\^L\relax%
1495 〈latexrelease〉\catcode`\^I\active%
1496 〈latexrelease〉
1497 〈latexrelease〉\gdef\filec@ntents#1{%
1498 〈latexrelease〉 \openin\@inputcheck#1 %
1499 〈latexrelease〉 \ifeof\@inputcheck%
1500 〈latexrelease〉 \ @latex@warning@no@line%
1501 〈latexrelease〉 \Writing file `@\currdir#1'}%
1502 〈latexrelease〉 \chardef\reserved@c15 %
1503 〈latexrelease〉 \ch@ck7\reserved@c\write%
1504 〈latexrelease〉 \immediate\openout\reserved@c#1\relax%
1505 〈latexrelease〉 \else%
1506 〈latexrelease〉 \closein\@inputcheck%
1507 〈latexrelease〉 \ @latex@warning@no@line%
1508 〈latexrelease〉 \ {File '#1' already exists on the system.\MessageBreak%
1509 〈latexrelease〉 \ Not generating it from this source}%
1510 〈latexrelease〉 \let\write\@gobbletwo%
1511 〈latexrelease〉 \let\closeout\@gobble%
1512 〈latexrelease〉 \fi%
1513 〈latexrelease〉 \if@tempswa%
1514 〈latexrelease〉 \immediate\write\reserved@c{%
1515 〈latexrelease〉 \ @percentchar\@percentchar\space%
1516 〈latexrelease〉 \expandafter\@gobble\string\LaTeXe file '#1'^^J%
1517 〈latexrelease〉 \ @percentchar\@percentchar\space generated by the %
1518 〈latexrelease〉 \ '@currenvir' \expandafter\@gobblefour\string\newenvironment^^J%
1519 〈latexrelease〉 \ @percentchar\@percentchar\space from source '\jobname' on %
1520 〈latexrelease〉 \number\year/\two@digits\month/\two@digits\day.^^J%
1521 〈latexrelease〉 \ @percentchar\@percentchar}%
1522 〈latexrelease〉 \fi%
1523 〈latexrelease〉 \let\do\@makeother\dospecials%
1524 〈latexrelease〉 \count@ 128\relax%
1525 〈latexrelease〉 \loop%
1526 〈latexrelease〉 \catcode\count@ 11\relax%
1527 〈latexrelease〉 \advance\count@ \@ne%
1528 〈latexrelease〉 \ifnum\count@<\@cclvi%
1529 〈latexrelease〉 \repeat%
1530 〈latexrelease〉 \edef\E{\@backslashchar end\string{\@currenvir\string}}%

```

```

1531 〈\latexrelease〉 \edef\reserved@b{%
1532 〈\latexrelease〉 \def\noexpand\reserved@b{%
1533 〈\latexrelease〉 #####1\E#####2\E#####3\relax}%
1534 〈\latexrelease〉 \reserved@b{%
1535 〈\latexrelease〉 \ifx\relax##3\relax%
1536 〈\latexrelease〉 \immediate\write\reserved@c{##1}%
1537 〈\latexrelease〉 \else%
1538 〈\latexrelease〉 \edef^~M{\noexpand\end{\currenvir}}%
1539 〈\latexrelease〉 \ifx\relax##1\relax%
1540 〈\latexrelease〉 \else%
1541 〈\latexrelease〉 \O@latex@warning{Writing text ‘##1’ before %
1542 〈\latexrelease〉 \string\end{\currenvir}\MessageBreak as last line of #1}%
1543 〈\latexrelease〉 \immediate\write\reserved@c{##1}%
1544 〈\latexrelease〉 \fi%
1545 〈\latexrelease〉 \ifx\relax##2\relax%
1546 〈\latexrelease〉 \else%
1547 〈\latexrelease〉 \O@latex@warning{%
1548 〈\latexrelease〉 Ignoring text ‘##2’ after \string\end{\currenvir}}%
1549 〈\latexrelease〉 \fi%
1550 〈\latexrelease〉 \fi%
1551 〈\latexrelease〉 ^~M}%
1552 〈\latexrelease〉 \catcode`^~L\active%
1553 〈\latexrelease〉 \let\^L\@undefined%
1554 〈\latexrelease〉 \def^~L{\expandafter\ifx\csname L\endcsname\relax\fi ^~J^~J}%
1555 〈\latexrelease〉 \catcode`^~I\active%
1556 〈\latexrelease〉 \let\^I\@undefined%
1557 〈\latexrelease〉 \def^~I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
1558 〈\latexrelease〉 \catcode`^~M\active%
1559 〈\latexrelease〉 \edef^~M##1^~M{%
1560 〈\latexrelease〉 \noexpand\reserved@b##1\^E\^E\relax}%
1561 〈\latexrelease〉 \endgroup%
1562 〈\latexrelease〉 \EndIncludeInRelease
1563 〈\latexrelease〉 {*2ekernel}
1564
1565 \begingroup
1566 \catcode`|= \catcode`\%
1567 \catcode`\#=12
1568 \catcode`*=11
1569 \gdef\@percentchar{%
1570 \gdef\endfilecontents{%
1571 \immediate\closeout\reserved@c
1572 \def\T##1##2##3{%
1573 \ifx##1\@undefined\else
1574 \O@latex@warning{no@line{##2 has been converted to Blank ##3e}%
1575 \fi}%
1576 \T\^L{Form Feed}{Lin}%
1577 \T\^I{Tab}{Spac}%
1578 \immediate\write\@unused{}}
1579 \global\let\endfilecontents*\endfilecontents

```

We no longer prevent the code to be used after begin document (no rollback needed for this change).

```

1580 %\onlypreamble\filecontents
1581 %\onlypreamble\endfilecontents

```

```

1582 %\@onlypreamble\filecontents*
1583 %\@onlypreamble\endfilecontents*
1584 \endgroup
1585 %\@onlypreamble\filecontents
(End of definition for \filecontents and \endfilecontents.)

```

## 5 Package/class rollback mechanism

```

1586 </2ekernel>
1587 (*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.

```

1588 (*tracer rollback)
1589 %\let\pkgcls@debug\typeout
1590 \let\pkgcls@debug\@gobble
1591 </tracer rollback>

```

(End of definition for \pkgcls@debug.)

\requestedLaTeXdate The macro (!) \requestedLaTeXdate holds the globally requested rollback date (via `latexrelease`) or zero if no such request was made.

```

1592 \def\requestedLaTeXdate{0}

```

(End of definition for \requestedLaTeXdate.)

\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 version 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.

```

1593 \ifx\pkgcls@targetdate\@undefined
1594 \newcount\pkgcls@targetdate
1595 \fi
1596 \let\pkgcls@targetlabel\@empty
1597 \def\pkgcls@innerdate{\maxdimen}

```

(End of definition for \pkgcls@targetdate, \pkgcls@targetlabel, and \pkgcls@innerdate.)

\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.

```

1598 \let\pkgcls@candidate\@empty
1599 \let\pkgcls@releasedate\@empty

```

(End of definition for \pkgcls@candidate and \pkgcls@releasedate.)

\load@onefilewithoptions the best place to add the rollback code is at the point where \onefilewithoptions is called to load a single class or package.

To make things easy we save the old definition as \load@onefilewithoptions and then provide a new interface.

Important: as this code is also unconditionally placed into latexrelease we can only do this name change once otherwise both macros will contain the same code.

```
1600 \ifx\load@onefilewithoptions\@undefined
1601 \let\load@onefilewithoptions\onefilewithoptions
1602 \def\onefilewithoptions#1[#2] [#3]#4{%
```

First a bit of tracing normally disabled.

```
1603 (*tracer rollback)
1604 \pkgcls@debug{--- File loaded request (\noexpand\usepackage or ...)}%
1605 \pkgcls@debug{\@spaces 1: #1}%
1606 \pkgcls@debug{\@spaces 2: #2}%
1607 \pkgcls@debug{\@spaces 3: #3}%
1608 \pkgcls@debug{\@spaces 4: #4}%
1609 (/tracer rollback)
```

Three of the arguments are needed later on in error/warning messages so we save them.

```
1610 \def\pkgcls@name{#1}%
1611 \def\pkgcls@arg {#3}%
1612 \edef\pkgcls@ext{%
1613 \ifx#4@\clsextension document class\else
1614 \ifx#4@\pkgextension package\else
1615 file
1616 \fi
1617 \fi
1618 }% % 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.

```
1619 \pkgcls@parse@date@arg{#3}%

```

When determining the correct release to load we keep track of candidates in \pkgcls@candidate and initially we don't have any:

```
1620 \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.

```
1621 \begingroup
1622 \edef\reserved@a{%
1623 \endgroup
1624 \unexpanded{\load@onefilewithoptions#1[#2]}%
1625 [\@nameuse{mindate}]%
1626 \unexpanded{#4}%
1627 \reserved@a
1628 }
1629 \fi
```

(End of definition for \load@onefilewithoptions and \onefilewithoptions.)

\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 that 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 \texrelease) 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    | \globalrollbackdate-as-number | \empty              | \empty          |
| \date     | \globalrollbackdate-as-number | \empty              | \date           |
| =\date    | \date-as-number               | \empty              | \empty          |
| =\version | 1                             | \version            | \empty          |
| \other    | \globalrollbackdate-as-number | \empty              | \other          |

where \globalrollbackdate-as-number is a date request given via \texrelease or if there wasn't one 0.

1630 \def\pkgcls@parse@date@arg #1{%

If the argument is empty we use the rollback date from \texrelease which has the value of zero if there was no rollback request. The label and the minimal date is made empty in that case.

1631 \ifx\@nil#1\@nil  
1632     \pkgcls@targetdate\requestedLaTeXdate\relax  
1633     \let\pkgcls@targetlabel\@empty  
1634     \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.

1635     \else  
1636         \pkgcls@parse@date@arg@#1=\@nil\relax  
1637     \fi  
1638 }

The actual parsing work then happens in \pkgcls@parse@date@arg@:

1639 \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.

1640     \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.

```
1641 \ifx\@nil#1\@nil
1642 \@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 =).

```
1643 \ifnum \pkgcls@targetdate=\z@
1644 \pkgcls@targetdate\@ne
1645 \def\pkgcls@innerdate{\maxdimen}%
1646 \pkgcls@parse@date@arg@version#2%
1647 \else
1648 \edef\pkgcls@innerdate{\the\pkgcls@targetdate}%
1649 \fi
1650 \let\pkgcls@mindate\@empty
1651 \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`):

```
1652 \requestedLaTeXdate\relax
1653 \let\pkgcls@targetlabel\@empty
1654 \def\pkgcls@innerdate{\maxdimen}%
1655 \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.

```
1656 \ifnum \pkgcls@targetdate > \z@
1657 \ifnum \@parse@version0#1//00\@nil > \pkgcls@targetdate
1658 \@latex@warning@no@line{Suspicious rollback/min-date date given\MessageBreak
1659 A minimal date of #1 has been specified for
1660 \pkgcls@ext\MessageBreak '\pkgcls@name'.\MessageBreak
1661 But this is in conflict
1662 with a rollback request to \requestedpatchdate}
1663 \fi
1664 \fi
1665 \fi
1666 }
```

Strip off the trailing = and assign the version name to `\pkgcls@targetlabel`.

```
1667 \def\pkgcls@parse@date@arg@version#1=%
1668 \def\pkgcls@targetlabel{\#1}}
```

*(End of definition for `\pkgcls@parse@date@arg`.)*

`\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.

```

1669 \def\DeclareRelease#1#2#3{%
1670 \ifnum\pkgcls@targetdate>\z@ % some sort of rollback request
1671 (*tracerollback)
1672 \pkgcls@debug{---\string\DeclareRelease:{}}
1673 \pkgcls@debug{\@spaces 1: #1}%
1674 \pkgcls@debug{\@spaces 2: #2}%
1675 \pkgcls@debug{\@spaces 3: #3}%
1676
```

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.

```

1677 \ifx\@nil#2\@nil
1678 \ifnum\pkgcls@targetdate=\@ne % named request
1679 \def\reserved@a{#1}%
1680 \ifx\pkgcls@targetlabel\reserved@a
1681 \pkgcls@use@this@release{#3}{}%
1682
```

```

1683 (*tracerollback)
1684 \else
1685 \pkgcls@debug{Label doesn't match}%
1686
```

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.

```
1693 \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.

```

1694 \ifnum\@parse@version#2//00\@nil
1695 >\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 when a package used didn't yet exists). So we make a complained to the user.

```

1696 \ifx\pkgcls@candidate\@empty
1697 \pkgcls@rollbackdate@error{#2}%
1698 \pkgcls@use@this@release{#3}{#2}%
1699
```

```

1700 \else
1701 \pkgcls@use@this@release\pkgcls@candidate
1702 \pkgcls@releasedate
1703
```

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.

```

1704 \def\pkgcls@candidate{\#3}%
1705 \def\pkgcls@releasedate{\#2}%
1706 <*tracer rollback>
1707 \pkgcls@debug{New candidate: #3}%
1708 </tracer rollback>
1709 \fi
1710 \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.

```

1711 \def\reserved@a{\#1}%
1712 \ifx\pkgcls@targetlabel\reserved@a
1713 \pkgcls@use@this@release{\#3}{\#2}%
1714 <*tracer rollback>
1715 \else
1716 \pkgcls@debug{Label doesn't match}%
1717 </tracer rollback>
1718 \fi
1719 \fi
1720 \fi
1721 \fi
1722 }

```

*(End of definition for \DeclareRelease.)*

`\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.

```
1723 \def\pkgcls@use@this@release#1#2{%
```

Before that we record the selection made inside the transcript.

```
1724 \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<sup>49</sup> and then we finally load the correct release.

After loading that file we need to stop reading the current file so we issue `\endinput`. Note that the `\relax` before that is essential to ensure that the `\endinput` is only happening after the file has been fully processed, otherwise it would act after the first line of the `\@@input!`

```

1725 \pkgcls@targetdate\z@
1726 \addtofilelist{\#1}%
1727 \@@input #1\relax
1728 \endinput
1729 }

```

*(End of definition for \pkgcls@use@this@release.)*

---

<sup>49</sup>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.

\pkgcls@show@selection This command records what selection was made. As that is needed in two places (and it is rather lengthy) it was placed in a separate command. The first argument is the name of the external file that is being loaded and is only needed for debugging. The second argument is the date that corresponds to this file and it is used as part of the message.

```

1730 \def\pkgcls@show@selection#1#2{%
1731 {*tracerollback}
1732 \pkgcls@debug{Result: use #1}%
1733 {/tracerollback}
1734 \GenericInfo
1735 {\@spaces\@spaces\space}{Rollback for
1736 \@cls@pkg\space'\@currname' requested ->
1737 \ifnum\pkgcls@targetdate>\@ne
1738 date
1739 \ifnum\requestedLaTeXdate=\pkgcls@targetdate
1740 \requestedpatchdate
1741 \else
1742 \expandafter\gobble\pkgcls@arg
1743 \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.

```

1744 Best approximation is
1745 \else
1746 version '\pkgcls@targetlabel'.\MessageBreak
1747 This corresponds to
1748 \fi
1749 \ifx\@nil#2\@nil
1750 a special release%
1751 \else
1752 the release introduced on #2%
1753 \fi
1754 \gobble}%
1755 }
```

(End of definition for \pkgcls@show@selection.)

\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.

```

1756 \def\pkgcls@rollbackdate@error#1{%
1757 @latex@error{Suspicious rollback date given}%
1758 {The \@cls@pkg\space'\@currname' has no rollback data
1759 before #1 which\MessageBreak
1760 is after your requested rollback date --- so
1761 something may be wrong here.\MessageBreak
1762 Continue and we use the earliest known release.}%

```

(End of definition for \pkgcls@rollbackdate@error.)

\DeclareCurrentRelease This declares the date (and possible name) of the current version of a package or class.

```

1763 \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).

```

1764 \ifnum\pkgcls@targetdate>\z0 % some sort of rollback request
1765 (*tracerrollback)
1766 \pkgcls@debug{---DeclareCurrentRelease}%
1767 \pkgcls@debug{ 1: #1}%
1768 \pkgcls@debug{ 2: #2}%
1769 (/tracerrollback)

```

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`.

```

1770 \ifnum\pkgcls@targetdate>\@ne % a date request
1771 \ifnum@\parse@version#2//00@nil
1772 >\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).

```

1773 \ifx\pkgcls@candidate@\empty
1774 \pkgcls@rollbackdate@error{#2}%
1775 \else
1776 \pkgcls@use@this@release\pkgcls@candidate
1777 \pkgcls@releasedate
1778 \fi

```

Otherwise the current file is the right release, so we record that in the transcript and then carry on.

```

1779 \else
1780 \pkgcls@show@selection{current version}{#2}%
1781 \fi
1782 \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.

```

1783 \def\reserved@a{#1}%
1784 \ifx\pkgcls@targetlabel\reserved@a
1785 \pkgcls@show@selection{current version}{#2}%
1786 \else
1787 \@latex@error{Requested version '\pkgcls@targetlabel' for
1788 '@cls@pkg@space' '@currname' is unknown}\@ehc
1789 \fi
1790 \fi
1791 \fi
1792 }

```

*(End of definition for `\DeclareCurrentRelease`.)*

- \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.

```
1793 \DeclareRobustCommand\IfTargetDateBefore[1]{%
1794 \ifnum\pkccls@innerdate <%
1795 \expandafter\@parse@version\expandafter0#1//00\@nil
1796 \typeout{Exclude code introduced on #1}%
1797 \expandafter\@firstoftwo
1798 \else
1799 \typeout{Include code introduced on #1}%
1800 \expandafter\@secondoftwo
1801 \fi
1802 }

(End of definition for \IfTargetDateBefore.)
```

```
1803 </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}`.

```
1804 <*afterpreamble>
1805 \NeedsTeXFormat{LaTeX2e}
1806 \ProvidesPackage{pkgindoc}
1807 [2020-08-08 v1.3m Package Interface in Document (DPC)]
1808 \def\reserved@a{\do\@classoptionslist\do\@filec\@ntents\relax}%
1809 \gdef\@preamblecmds{\#1\#3}%
1810 \expandafter\reserved@a\@preamblecmds\relax
1811 </afterpreamble>
```

# File 51

## ltkeys.dtx

### 1 Creating and using keyval options

As with any key–value input, using key–value pairs as package or class options has two parts: creating the key options and setting (using) them. Options created in this way *may* be used after package loading as general key–value settings: this will depend on the nature of the underlying code.

---

```
\DeclareKeys \DeclareKeys [{family}] {declarations}
```

Creates a series of options from a comma-separated *declarations* list. Each entry in this list is a key–value pair, with the *key* having one or more *properties*. A small number of “basic” *properties* are described below. The full range of properties, provided by *l3keys*, can also be used for more powerful processing. See *interface3* for the full details.

The basic properties provided here are

- **.code** — execute arbitrary code
- **.if** — sets a *TeX* `\if...` switch
- **.ifnot** — sets an inverted *TeX* `\if...` switch
- **.pass-to-packages** — for class options, this specifies whether the option should be treated “global” (read by packages from the global list); for package options this property has no effect
- **.store** — stores a value in a macro
- **.usage** — defines whether the option can be given only when loading (*load*), in the preamble (*preamble*) or has no limitation on scope (*general*)

The part of the *key* before the *property* is the *name*, with the *value* working with the *property* to define the behaviour of the option.

For example, with

```
\DeclareKeys[mypkg]
{
 draft.if = @mypkg@draft ,
 draft.usage = preamble ,
 name.store = \@mypkg@name ,
 name.usage = load ,
 second-name.store = \@mypkg@other@name
}
```

three options would be created. The option `draft` can be given anywhere in the preamble, and will set a switch called `\if@mypkg@draft`. The option `name` can only be given during package loading, and will save whatever value it is given in `\@mypkg@name`. Finally, the option `second-name` can be given anywhere, and will save its value in `\@mypkg@other@name`.

Keys created *before* the use of `\ProcessKeyOptions` act as package options.

---

```
\DeclareUnknownKeyHandler \DeclareUnknownKeyHandler [<family>] {{code}}
```

The function `\DeclareUnknownKeyHandler` may be used to define the behavior when an undefined key is encountered. The `<code>` will receive the unknown key name as #1 and the value as #2. These can then be processed as appropriate, e.g. by forwarding to another package.

---

```
\ProcessKeyOptions \ProcessKeyOptions [<family>]
```

The `\ProcessKeyOptions` function is used to check the current option list against the keys defined for `<family>`. Global (class) options and local (package) options are checked when this function is called in a package.

---

```
\SetKeys \SetKeys [<family>] {{keyvals}}
```

Sets (applies) the explicit list of `<keyvals>` for the `<family>`: if the latter is not given, the value of `\@currname` used. This command may be used within a package to set options before or after using `\ProcessKeyOptions`.

## 1.1 Implementation of `lkeys`

```
1 (@=keys)
2 (*2ekernel)
3 \ExplSyntaxOn
```

## 1.2 Key properties

```
.code
.if
4 \group_begin:
.ifnot
5 \cs_set_protected:Npn __keys_tmp:nn #1#2
.store
6 {
.usage
7 \quark_if_recursion_tail_stop:n {#1}
8 \cs_new_eq:cc
9 { \c_keys_props_root_str . #2 }
10 { \c_keys_props_root_str . #1 }
11 __keys_tmp:nn
12 }
13 __keys_tmp:nn
14 { code:n } { code }
15 { legacy_if_set:n } { if }
16 { legacy_if_set_inverse:n } { ifnot }
17 { tl_set:N } { store }
18 { usage:n } { usage }
19 { \q_recursion_tail } { }
20 \q_recursion_stop
21 \group_end:
```

(End of definition for `.code` and others.)

`.pass-to-packages` Used to force options to be global: as this property (uniquely) has an *optional* value, there is a bit of work to do.

```
__keys_scope:n
__keys_scope:N
22 \cs_new_protected:cpn { \c_keys_props_root_str .pass-to-packages }
23 {
24 \bool_if:NTF \l__keys_no_value_bool
```

```

25 { __keys_scope:n { true } }
26 { __keys_scope:n }
27 }
28 \cs_new_protected:Npn __keys_scope:n #1
29 {
30 \str_case:nnF {#1}
31 {
32 { true }
33 { __keys_scope:N \clist_put_right:NV }
34 { false }
35 { __keys_scope:N \clist_remove_all:NV }
36 }
37 {
38 \msg_error:nnnn { keys }
39 { choice-unknown }
40 { .pass-to-packages }
41 {#1}
42 }
43 }
44 \cs_new_protected:Npn __keys_scope:N #1
45 {
46 \exp_after:wN __keys_find_key_module:wNN
47 \l_keys_path_str \s__keys_stop
48 \l_keys_key_tl \l_keys_key_str
49 #1 \l__keys_forced_global_clist \l_keys_key_str
50 }

```

(End of definition for `.pass-to-packages`, `\__keys_scope:n`, and `\__keys_scope:N`.)

### 1.3 Main mechanism

```

51 \cs_generate_variant:Nn \clist_if_in:NnT { Ne }
52 \cs_generate_variant:Nn \clist_if_in:NnTF { Ne }
53 \cs_generate_variant:Nn \clist_put_right:Nn { Nv }

```

`\l__keys_class_only_clist` Used to track class-only options.

```
54 \clist_new:N \l__keys_class_only_clist
```

(End of definition for `\l__keys_class_only_clist`.)

`\l__keys_forced_global_clist` Used to force options to be global.

```
55 \clist_new:N \l__keys_forced_global_clist
```

(End of definition for `\l__keys_forced_global_clist`.)

`\l__keys_options_clist` A single list is used for all options, into which they are collected before processing.

```
56 \clist_new:N \l__keys_options_clist
```

(End of definition for `\l__keys_options_clist`.)

---

### `\l__keys_options_loading_bool`

Used to indicate we are in the loading phase: controls the outcome of warnings.

```
57 \bool_new:N \l__keys_options_loading_bool
```

`\__keys_options:n` The main function calls functions to collect up the global and local options into `\l__keys_options_clist` before calling the underlying functions to actually do the processing. So that a suitable message is produced if the option is unknown, the special `unknown` key is set if it does not already exist for the current family, and is cleaned up afterwards if required. To allow the L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  layer to know this mechanism is active, and to deal with the key family not matching the file name, we store the family in all cases.

```

58 \cs_new_protected:Npn __keys_options:n #1
59 { __keys_options_expand_module:Nn __keys_options_aux:n {#1} }
60 \cs_new_protected:Npn __keys_options_aux:n #1
61 {
62 \cs_gset_protected:cpn { opt@handler@\currname.\current }
63 { \ProcessKeyOptions [#1] }
64 \cs_set_protected:Npn __keys_option_end: { }
65 \clist_clear:N \l__keys_options_clist
66 __keys_options_global:n {#1}
67 __keys_options_local:
68 \keys_if_exist:nnF {#1} { unknown }
69 {
70 \keys_define:nn {#1}
71 {
72 unknown .code:n =
73 {
74 \msg_error:nnxx { keys } { option-unknown }
75 { \l_keys_key_str } { \currname }
76 }
77 }
78 \cs_set_protected:Npn __keys_option_end:
79 { \keys_define:nn {#1} { unknown .undefine: } }
80 }
81 \bool_set_true:N \l__keys_options_loading_bool
82 \clist_map_variable:NNn \l__keys_options_clist \CurrentOption
83 { \keys_set:nV {#1} \CurrentOption }
84 \bool_set_false:N \l__keys_options_loading_bool
85 \AtEndOfPackage { \cs_set_eq:NN \unprocessedoptions \scan_stop: }
86 __keys_option_end:
87 __keys_options_loaded:n {#1}
88 }

89 \msg_new:nnnn { keys } { option-unknown }
90 { Unknown-option-'#1'-for-package-'#2'. }
91 {
92 LaTeX-has-been-asked-to-set-an-option-called-'#1'-
93 but-the-package-"\msg_module_name:n {#2}"-has-not-created-an-option-with-this-name.
94 }

```

(End of definition for `\__keys_options:n`, `\__keys_options_aux:n`, and `\__keys_options_end:..`)

`\__keys_options_global:n` Global (class) options are handled differently for L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  packages and classes. Hence this function is essentially a check on the current file type. The initial test is needed as L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  allows variables to be equal to `\scan_stop:`, which is usually forbidden in `expl3` code.

```

95 \cs_new_protected:Npn __keys_options_global:n #1
96 {
97 \cs_if_eq:NNF \raw@classoptionslist \scan_stop:

```

```

98 {
99 \cs_if_eq:NNTF \currext \clsextension
100 { __keys_options_class:n {#1} }
101 { __keys_options_package:n {#1} }
102 }
103 }

```

*(End of definition for \\_\_keys\_options\_global:n.)*

\\_\_keys\_options\_class:n For classes, each option (stripped of any content after =) is checked for existence as a key. If found, the option is added to the combined list for processing. On the other hand, unused options are stored up in \unusedoptionlist. An earlier version of this code checked for the `unknown` key just once and if found short-cutted the loop: that though makes handling more complex situations harder, so we take the performance hit instead. Options used by classes are tracked but the catch-all `unknown` is excluded (hence not using a lazy evaluation for the key testing).

```

104 \cs_new_protected:Npn __keys_options_class:n #1
105 {
106 \cs_if_free:cF { @raw@opt@ \currname . \currext }
107 {
108 \clist_map_inline:cn { @raw@opt@ \currname . \currext }
109 {
110 \exp_args:Ne __keys_options_class:nnn
111 { \tl_trim_spaces:e { __keys_remove_equals:n {##1} } }
112 {##1} {#1}
113 }
114 }
115 }
116 \cs_new_protected:Npn __keys_options_class:nnn #1#2#3
117 {
118 \keys_if_exist:nnTF {#3} {#1}
119 {
120 __keys_options_class:nn {#1} {#2}
121 \clist_put_right:Ne \l__keys_class_only_clist { \tl_to_str:n {#1} }
122 }
123 {
124 \keys_if_exist:nnTF {#3} { unknown }
125 { __keys_options_class:nn {#1} {#2} }
126 {
127 \clist_if_in:NnF \unusedoptionlist {#1}
128 { \clist_put_right:Nn \unusedoptionlist {#1} }
129 }
130 }
131 }
132 \cs_new_protected:Npn __keys_options_class:nn #1#2
133 {
134 \clist_remove_all:Nn \unusedoptionlist {#1}
135 \clist_put_right:Nn \l__keys_options_clist {#2}
136 }

```

*(End of definition for \\_\_keys\_options\_class:n, \\_\_keys\_options\_class:nn, and \\_\_keys\_options\_class:nn.)*

`\__keys_options_package:n` For global options when processing a package, the tasks are slightly different from those for a class. The check is the same, but here there is nothing to do if the option is not applicable. Each valid option also needs to be removed from `\@unusedoptionlist`.

```

137 \cs_new_protected:Npn __keys_options_package:n #1
138 {
139 \clist_map_inline:Nn \@raw@classoptionslist
140 {
141 \exp_args:Ne __keys_options_package:nnn
142 { \tl_trim_spaces:e { __keys_remove_equals:n {##1} } }
143 {##1} {#1}
144 }
145 }

```

The forced-global test here needs to use `\tl_to_str:n` as the data come from a key name, which is always a string.

```

146 \cs_new_protected:Npn __keys_options_package:nnn #1#2#3
147 {
148 \keys_if_exist:nnT {#3} {#1}
149 {
150 \clist_if_in:NeTF \l__keys_class_only_clist { \tl_to_str:n {#1} }
151 {
152 \clist_if_in:NeT \l__keys_forced_global_clist { \tl_to_str:n {#1} }
153 { __keys_options_package:nn {#1} {#2} }
154 }
155 { __keys_options_package:nn {#1} {#2} }
156 }
157 }
158 \cs_new_protected:Npn __keys_options_package:nn #1#2
159 {
160 \clist_put_right:Nn \l__keys_options_clist {#2}
161 \clist_remove_all:Nn \@unusedoptionlist {#1}
162 }

```

(End of definition for `\__keys_options_package:n`, `\__keys_options_package:nnn`, and `\__keys_options_package:nn`.)

`\__keys_options_local:` If local options are found, they are added to the processing list. L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> stores options for each file in a macro which may or may not exist, hence the need to use `\cs_if_exist:c`.

```

163 \cs_new_protected:Npn __keys_options_local:
164 {
165 \cs_if_eq:NNF \@currext \@clsextension
166 {
167 \cs_if_exist:cT { \raw@opt@ \currname . \@currext }
168 {
169 \clist_put_right:Nv \l__keys_options_clist
170 { \raw@opt@ \currname . \@currext }
171 }
172 }
173 }

```

(End of definition for `\__keys_options_local:)`)

`\__keys_remove_equals:n` As the name suggests, this is a simple function to remove an equals sign from the input.  
`\__keys_remove_equals:w` This is all wrapped up in an `n` function so that there will always be a sign available.

```

174 \cs_new:Npn __keys_remove_equals:n #1
175 { __keys_remove_equals:w #1 = \s__keys_stop }
176 \cs_new:Npn __keys_remove_equals:w #1 = #2 \s__keys_stop { \exp_not:n {#1} }

```

(End of definition for `\__keys_remove_equals:n` and `\__keys_remove_equals:w`.)

## 1.4 The document interfaces

```

177 \cs_generate_variant:Nn \keys_define:nn { nx }

```

`\__keys_options_expand_module:Nn`  
`\__keys_options_expand_module:nN`

To deal with active characters inside the module argument whilst also expanding that argument, we use a combination of c- and f-type expansion. This works as the definitions for active UTF-8 bytes contain an `\ifincharname` test.

```

178 \cs_new_protected:Npn __keys_options_expand_module:Nn #1#2
179 {
180 \cs:w __keys_options_expand_module:nN \use:e { \cs_end: {#2} } #1
181 }
182 \cs_new_protected:Npn __keys_options_expand_module:nN #1#2
183 { #2 {#1} }

```

(End of definition for `\__keys_options_expand_module:Nn` and `\__keys_options_expand_module:nN`.)

### \DeclareKeys

Defining key options is quite straight-forward: we have an intermediate function to allow for potential set-up steps.

```

184 \NewDocumentCommand \DeclareKeys { O { \currname } +m }
185 { __keys_options_expand_module:Nn \keys_define:nn {#1} {#2} }

```

(End of definition for `\DeclareKeys`. This function is documented on page 1027.)

### \DeclareUnknownKeyHandler

```

186 \NewDocumentCommand \DeclareUnknownKeyHandler { O { \currname } +m }
187 {
188 \cs_set_protected:cpn { __keys_unknown_handler_ #1 :nn } ##1##2 {#2}
189 __keys_options_expand_module:Nn \keys_define:nx {#1}
190 {
191 unknown .code:n =
192 \exp_not:N \exp_args:NV
193 \exp_not:c { __keys_unknown_handler_ #1 :nn }
194 \exp_not:N \l_keys_key_str {####1}
195 }
196 }

```

(End of definition for `\DeclareUnknownKeyHandler`. This function is documented on page 1028.)

### \ProcessKeyOptions

We need to deal with the older interface from l3keys2e here: it had a mandatory argument. We can mop that up using a look-ahead, and then exploit that information to determine whether the package option handling is set up for the new approach for clash handling.

```

197 \NewDocumentCommand \ProcessKeyOptions { O { \currname } }
198 { __keys_options:n {#1} }
199 \onlypreamble \ProcessKeyOptions

```

(End of definition for `\ProcessKeyOptions`. This function is documented on page 1028.)

## 1.5 Option usage scope

`\__keys_options_loaded:n` Indicates that the load-time options for a package have been processed: once this has happened, make them unavailable either with a warning or an error.

```

200 \cs_new_protected:Npn __keys_options_loaded:n #1
201 {
202 \prop_get:NnNT \l_keys_usage_load_prop {#1} \l__keys_tmpa_tl
203 {
204 \clist_map_inline:Nn \l__keys_tmpa_tl
205 {
206 \keys_define:nn {#1}
207 {
208 ##1 .code:n =
209 __keys_options_loaded:nn {#1} {##1}
210 }
211 }
212 }
213 }
214 \cs_new_protected:Npn __keys_options_loaded:nn #1#2
215 {
216 \bool_if:NTF \l__keys_options_loading_bool
217 { \msg_warning:nnnn { keys } { load-option-ignored } }
218 { \msg_error:nnnn { keys } { load-only } }
219 {#1} {#2}
220 }

221 \msg_new:nnn { keys } { load-option-ignored }
222 {
223 Package~"\msg_module_name:n {#1}"~has~already~been~loaded:~
224 ignoring~load-time~option~"#2".
225 }

226 \msg_new:nnnn { keys } { load-only }
227 {
228 Key~"#2"~may~only~be~used~during~loading~of~package~
229 "\msg_module_name:n {#1}".
230 }
231 {
232 LaTeX-was~asked~to~set~a~key~called~"#2",~but~this~is~only~allowed~
233 in~the~optional~argument~when~loading~package~"\msg_module_name:n{#1}".
234 }
```

(End of definition for `\__keys_options_loaded:n` and `\__keys_options_loaded:nn`.)

Disable all preamble options in one shot.

```

235 \tl_gput_left:Nn \c_kernel@after\begindocument
236 {
237 \prop_map_inline:Nn \l_keys_usage_preamble_prop
238 {
239 \clist_map_inline:nn {#2}
240 {
241 \keys_define:nn {#1}
242 {
243 ##1 .code:n =
244 \msg_error:nnn { keys } { preamble-only } {##1}
245 }
246 }
247 }
248 }
```

```

246 }
247 }
248 }
249 \msg_new:nnnn { keys } { preamble-only }
250 { Key~"#1"~may~only~be~used~in~the~preamble. }
251 {
252 LaTeX~was~asked~to~set~a~key~called~"#1",~but~this~is~only~allowed~
253 before~\begin{document}.~You~will~need~to~set~the~key~earlier.
254 }
```

## 1.6 General key setting

**\SetKeys** A simple wrapper.

```

255 \NewDocumentCommand \SetKeys { O { \currname } +m }
256 { \keys_options_expand_module:Nn \keys_set:nn {#1} {#2} }
```

(End of definition for **\SetKeys**. This function is documented on page 1028.)

```

257 \ExplSyntaxOff
258
```

# File 52

## ltfilehook.dtx

### 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 `\Cinput` or `\openin`.

---

`file/before`  
`file/.../before`  
`file/.../after`  
`file/after`

---

These are:

`file/before`, `file/<file-name>/before` 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/<file-name>/after`, `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 L<sup>A</sup>T<sub>E</sub>X is:

```
\UseHook{\file/before}
\UseHook{\file/<file name>/before}
 <file contents>
\UseHook{\file/<file name>/after}
\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  $\text{\TeX}$  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/\{file name\}/before` and `file/\{file name\}/after`.

---

`\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  $\text{\TeX}$  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  $\text{\TeX}$ , 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/array.sty/after` 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` These are:  
`package/after`  
`package/.../before` **package/before, package/after** These hooks are called for each package being loaded.  
`package/.../after`  
`class/before` **package/\{name\}/before, package/\{name\}/after** These hooks are additionally called if the package name is `\{name\}` (without extension).  
`class/after`  
`class/.../before`  
`class/.../after` **class/before, class/after** These hooks are called for each class being loaded.  
`class/\{name\}/before` **class/\{name\}/after** 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>/\<package name>/before}
 \UseHook{\<file>/before}
 \UseHook{\<file>/\<package name>.sty/before}
 <package contents>
 \UseHook{\<file>/\<package name>.sty/after}
 \UseHook{\<file>/after}

code from \AtEndOfPackage if used inside the package

\UseHook{\<package>/\<package name>/after}
\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!

All class or package hooks involving the name of the class or package are implemented as one-time hooks, whereas all other such hooks are normal hooks. This allows for the following use case

```
\AddToHook{package/varioref/after}
 { ... apply my customizations if the package gets
 loaded (or was loaded already) ... }
```

without the need to first test if the package is already loaded.

## 1.4 Hooks for `\include` files

To manage `\include` files, L<sup>A</sup>T<sub>E</sub>X 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 `\clearpages` 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.<sup>50</sup> None of the hooks are executed when an `\include` file is bypassed because of an `\includeonly` declaration. They are, however, all executed if L<sup>A</sup>T<sub>E</sub>X makes an attempt to load the `\include` file (even if it doesn't exist and all that happens is "No file `\<filename>.tex`").

---

<sup>50</sup>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/<name>/before` 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/<name>/end, include/end` These hooks are executed (in that order) after L<sup>A</sup>T<sub>E</sub>X has stopped reading from the `\include` file, but before it has issued a `\clearpage` to output any deferred floats.

`include/<name>/after, include/after` These hooks are executed (in that order) after L<sup>A</sup>T<sub>E</sub>X has issued the `\clearpage` but before it 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.<sup>51</sup>

`include/excluded, include/<name>/excluded` The above hooks for `\include` files are only executed when the file is loaded (or more exactly the load is attempted). If, however, the `\include` file is explicitly excluded (through an `\includeonly` statement) the above hooks are bypassed and instead the `include/excluded` hook followed by the `include/<name>/excluded` hook are executed. This happens after L<sup>A</sup>T<sub>E</sub>X has loaded the `.aux` file for this include file, i.e., after L<sup>A</sup>T<sub>E</sub>X has updated its counters to pretend that the file was seen.

All `include` hooks involving the name of the included file are implemented as one-time hooks (whereas all other such hooks are normal hooks).

If you want to execute code that is run for every `\include` regardless of whether or not it is excluded, use the `cmd/include/before` or `cmd/include/after` hooks.

## 1.5 High-level interfaces for L<sup>A</sup>T<sub>E</sub>X

We do not provide any additional wrappers around the hooks (like `filehook` or `scrlfile`) because we believe that for package writers the high-level commands from the hook management, e.g., `\AddToHook`, etc. are sufficient and in fact easier to work with, given that the hooks have consistent naming conventions.

---

<sup>51</sup>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.

## 1.6 Kernel, class, and package interfaces for L<sup>A</sup>T<sub>E</sub>X

---

```
\declare@file@substitution \declare@file@substitution {\langle file\rangle} {\langle replacement-file\rangle}
\undeclare@file@substitution \undeclare@file@substitution {\langle file\rangle}
```

---

If  $\langle file \rangle$  is requested for loading replace it with  $\langle replacement-file \rangle$ .  $\CurrentFile$  remains pointing to  $\langle file \rangle$  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 L<sup>A</sup>T<sub>E</sub>X 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. As such it is mainly meant for use in the L<sup>A</sup>T<sub>E</sub>X kernel but other use cases are conceivable.

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 implementing the same functionality as the original one!*

---

```
\disable@package@load \disable@package@load {\langle package\rangle} {\langle alternate-code\rangle}
\reenable@package@load \reenable@package@load {\langle package\rangle}
```

---

If  $\langle package \rangle$  is requested, do not load it but instead run  $\langle alternate-code \rangle$  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 and again it should only be applied if there are good reasons for doing this.<sup>52</sup>

## 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 as 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
```

---

<sup>52</sup>Just to be sure: "I don't like this package by somebody else" is not a good one :-)

```

= (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

```

1 {*2ekernel}
2 @=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 L<sup>A</sup>T<sub>E</sub>X, whereas the other two hold the name as requested. They will differ in case there's a file substitution.

```

3 </2ekernel>
4 {*2ekernel | latexrelease}
5 (latexrelease)\IncludeInRelease{2020/10/01}%
6 (latexrelease) {\CurrentFile}{Hook management file}%
7 \ExplSyntaxOn
8 \tl_new:N \CurrentFile
9 \tl_new:N \CurrentFilePath
10 \tl_new:N \CurrentFileUsed
11 \tl_new:N \CurrentFilePathUsed

```

```

12 \ExplSyntaxOff
13 </2ekernel | latexrelease>
14 <latexrelease>\EndIncludeInRelease
15 <latexrelease>\IncludeInRelease{0000/00/00}%
16 <latexrelease> {\CurrentFile}{Hook management file}%
17 <latexrelease>
18 <latexrelease>\let \CurrentFile \@undefined
19 <latexrelease>\let \CurrentFilePath \@undefined
20 <latexrelease>\let \CurrentFileUsed \@undefined
21 <latexrelease>\let \CurrentFilePathUsed \@undefined
22 <latexrelease>
23 <latexrelease>\EndIncludeInRelease
24 <*2ekernel>

```

(End of definition for `\CurrentFile` and others. These functions are documented on page 1037.)

## 2.2 `expl3` helpers

```

25 </2ekernel>
26 <*2ekernel | latexrelease>
27 <latexrelease>\IncludeInRelease{2020/10/01}%
28 <latexrelease> {__filehook_file_parse_full_name:nN}{File helpers}%
29 \ExplSyntaxOn

```

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 normalization 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.

```

30 \cs_new:Npn __filehook_file_parse_full_name:nN #1
31 {
32 \exp_args:Nf \file_parse_full_name_apply:nN
33 {
34 \exp_args:Nf __filehook_full_name:nn
35 { \file_full_name:n {#1} } {#1}
36 }
37 }
38 \cs_new:Npn __filehook_full_name:nn #1 #2
39 {
40 \tl_if_empty:nTF {#1}
41 { \tl_trim_spaces:n {#2} }
42 { \tl_trim_spaces:n {#1} }
43 }

```

(End of definition for `\_\_filehook_file_parse_full_name:nN` and `\_\_filehook_full_name:nn`.)

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.

```

44 \cs_new:Npn __filehook_if_no_extension:nTF #1
45 {
46 \exp_args:Ne \tl_if_empty:nTF
47 { \file_parse_full_name_apply:nN {#1} \use_iii:nnn }
48 }

```

```

49 \cs_new_protected:Npn __filehook_drop_extension:N #1
50 {
51 \tl_gset:Nx #1
52 {
53 \exp_args:NV __filehook_file_parse_full_name:nN #1
54 __filehook_drop_extension_aux:nnn
55 }
56 }
57 \cs_new:Npn __filehook_drop_extension_aux:nnn #1 #2 #3
58 { \tl_if_empty:nF {#1} { #1 / } #2 }

(End of 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.

```

59 \tl_new:N \l__filehook_internal_tl
60 \seq_if_exist:NF \g__filehook_input_file_seq
61 { \seq_new:N \g__filehook_input_file_seq }
62 \cs_new_protected:Npn __filehook_file_push:
63 {
64 \seq_gpush:Nx \g__filehook_input_file_seq
65 {
66 { \CurrentFilePathUsed } { \CurrentFileUsed }
67 { \CurrentFilePath } { \CurrentFile }
68 }
69 }
70 \cs_new_protected:Npn __filehook_file_pop:
71 {
72 \seq_gpop:NNTF \g__filehook_input_file_seq \l__filehook_internal_tl
73 { \exp_after:wN __filehook_file_pop_assign:nnnn \l__filehook_internal_tl }
74 {
75 \msg_error:nnn { latex2e } { should-not-happen }
76 { Tried-to-pop~from~an~empty~file~name~stack. }
77 }
78 }
79 \cs_new_protected:Npn __filehook_file_pop_assign:nnnn #1 #2 #3 #4
80 {
81 \tl_set:Nn \CurrentFilePathUsed {#1}
82 \tl_set:Nn \CurrentFileUsed {#2}
83 \tl_set:Nn \CurrentFilePath {#3}
84 \tl_set:Nn \CurrentFile {#4}
85 }
86 \ExplSyntaxOff
```

(End of definition for `\g__filehook_input_file_seq` and others.)

```
87 ⟨/2ekernel | latexrelease⟩
88 ⟨latexrelease⟩\EndIncludeInRelease
```

When rolling forward the following `expl3` functions may not be defined. If we roll back the code does nothing.

```

89 〈latexrelease〉\IncludeInRelease{2020/10/01}%
90 〈latexrelease〉 {\file_parse_full_name_apply:nN}{Roll forward help}%
91 〈latexrelease〉
92 〈latexrelease〉\ExplSyntaxOn
93 〈latexrelease〉\cs_if_exist:NF\file_parse_full_name_apply:nN
94 〈latexrelease〉{
95 〈latexrelease〉\cs_new:Npn \file_parse_full_name_apply:nN #1
96 〈latexrelease〉 {
97 〈latexrelease〉 \exp_args:Ne __file_parse_full_name_auxi:nN
98 〈latexrelease〉 { __kernel_file_name_sanitize:n {#1} }
99 〈latexrelease〉 }
100 〈latexrelease〉\cs_new:Npn __file_parse_full_name_auxi:nN #1
101 〈latexrelease〉 {
102 〈latexrelease〉 __file_parse_full_name_area:nw { } #1
103 〈latexrelease〉 / \s__file_stop
104 〈latexrelease〉 }
105 〈latexrelease〉\cs_new:Npn __file_parse_full_name_area:nw #1 #2 / #3 \s__file_stop
106 〈latexrelease〉 {
107 〈latexrelease〉 \tl_if_empty:nTF {#3}
108 〈latexrelease〉 { __file_parse_full_name_base:nw { } #2 . \s__file_stop {#1} }
109 〈latexrelease〉 { __file_parse_full_name_area:nw { #1 / #2 }
110 〈latexrelease〉 #3 \s__file_stop }
111 〈latexrelease〉 }
112 〈latexrelease〉\cs_new:Npn __file_parse_full_name_base:nw #1 #2 . #3 \s__file_stop
113 〈latexrelease〉 {
114 〈latexrelease〉 \tl_if_empty:nTF {#3}
115 〈latexrelease〉 {
116 〈latexrelease〉 \tl_if_empty:nTF {#1}
117 〈latexrelease〉 {
118 〈latexrelease〉 \tl_if_empty:nTF {#2}
119 〈latexrelease〉 { __file_parse_full_name_tidy:nnnN { } { } }
120 〈latexrelease〉 { __file_parse_full_name_tidy:nnnN { .#2 } { } }
121 〈latexrelease〉 }
122 〈latexrelease〉 { __file_parse_full_name_tidy:nnnN {#1} { .#2 } }
123 〈latexrelease〉 }
124 〈latexrelease〉 { __file_parse_full_name_base:nw { #1 . #2 }
125 〈latexrelease〉 #3 \s__file_stop }
126 〈latexrelease〉 }
127 〈latexrelease〉\cs_new:Npn __file_parse_full_name_tidy:nnnN #1 #2 #3 #4
128 〈latexrelease〉 {
129 〈latexrelease〉 \exp_args:Nee #4
130 〈latexrelease〉 {
131 〈latexrelease〉 \str_if_eq:nnF {#3} { / } { \use_none:n }
132 〈latexrelease〉 #3 \prg_do_nothing:
133 〈latexrelease〉 }
134 〈latexrelease〉 { \use_none:n #1 \prg_do_nothing: }
135 〈latexrelease〉 {#2}
136 〈latexrelease〉 }
137 〈latexrelease〉
138 〈latexrelease〉\ExplSyntaxOff
139 〈latexrelease〉
140 〈latexrelease〉\EndIncludeInRelease
141 〈*2ekernel〉
142 〈@@=〉

```

## 2.3 Declaring the file-related hooks

These hooks have names with three-parts that start with `file/`, `include/`, `class/` or `package/` and end with `/before` or `/after` (or `/end` in the case of `include/`). They are all generic hooks so will be declared only if code is added to them; this declaration is done for you automatically and, indeed, they should not be declared explicitly.

Those named `.../after` and `include/.../end` are, when code is added, declared as reversed hooks.

## 2.4 Patching L<sup>A</sup>T<sub>E</sub>X's \InputIfFileExists command

Most of what we have to do is adding `\UseHook` into several L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  core commands, because of some circular dependencies in the kernel we do this only now and not in `ltfiles`.

```
\InputIfFileExists \InputIfFileExists loads any file if it is available so we have to add the hooks
@input@file@exists@with@hooks file/before and file/after in the right places. If the file doesn't exist no hooks
\unqu@tefilef@und should be executed.
```

```
143 </2ekernel>
144 <latexrelease>\IncludeInRelease{2020/10/01}%
145 <latexrelease> {\InputIfFileExists}{Hook management (files)}%
146 <*2ekernel | latexrelease>

147 \let\InputIfFileExists@\undefined
148 \DeclareRobustCommand \InputIfFileExists[2]{%
149 \IfFileExists{#1}%
150 {%
151 \@expl@@@filehook@file@push@@
152 \@filehook@set@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.

```
153 \expandafter\@swaptwoargs\expandafter
154 {\expandafter\@input@file@exists@with@hooks
155 \expandafter{\@filef@und}}%
156 {#2}%
157 \@expl@@@filehook@file@pop@@
158 }%
159 }
160 \def\@input@file@exists@with@hooks#1{%
```

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`.

```

161 \edef\reserved@a{%
162 \@expl@@@filehook@file@pop@assign@nnnn
163 {\CurrentFilePathUsed}%
164 {\CurrentFileUsed}%
165 {\CurrentFilePath}%
166 {\CurrentFile}}%
167 \expandafter\swaptwoargs\expandafter{\reserved@a}%

```

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).

```

168 {\edef\reserved@a{\unqu@tefilef@und#1\@nil}%
169 \@addtofilelist{\string\makeletter\reserved@a}%
170 \UseHook{file/before}}%

```

The current file name is available in `\CurrentFile` so we use that in the specific hook.

```

171 \UseHook{file/\CurrentFile/before}%
172 \@@input #1% <- trailing space comes from \@filef@und
173 }%

```

And here, `\CurrentFile` is restored (by `\@expl@@@filehook@file@pop@assign@nnnn`) so we can use it once more.

```

174 \UseHook{file/\CurrentFile/after}%
175 \UseHook{file/after}%
176 \def\unqu@tefilef@und"#1" \@nil{#1}

```

Now declare the non-generic file hooks used above:

```

177 \NewHook{file/before}
178 \NewReversedHook{file/after}
179 {latexrelease}\EndIncludeInRelease
180 {/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'.

```

181 {latexrelease}\IncludeInRelease{2019/10/01}%
182 {latexrelease} {\InputIfFileExists}{Hook management (files)}%
183 {latexrelease}%
184 {latexrelease}\DeclareRobustCommand \InputIfFileExists[2]{%
185 {latexrelease} \IfFileExists{#1}%
186 {latexrelease} {%

```

```

187 〈\latexrelease〉 \expandafter\@swaptwoargs\expandafter
188 〈\latexrelease〉 { \@filef@und } { \#2\@addtofilelist{#1}\@@input } } }
189 〈\latexrelease〉 \let\@input@file@exists@with@hooks\@undefined
190 〈\latexrelease〉 \let\unqu@tefilef@und\@undefined
191 〈\latexrelease〉 \EndIncludeInRelease
192 〈\latexrelease〉 \IncludeInRelease{0000/00/00}%
193 〈\latexrelease〉 { \InputIfFileExists } { Hook management (files) } %
194 〈\latexrelease〉 \long\def \InputIfFileExists#1#2{%
195 〈\latexrelease〉 \IfFileExists{#1}%
196 〈\latexrelease〉 { \#2\@addtofilelist{#1}\@@input \@filef@und } }

```

Also undo the internal command as some packages unfortunately test for their existence instead of using \IfFormatAtLeastTF.

```

197 〈\latexrelease〉 \expandafter\let\csname InputIfFileExists \endcsname\@undefined
198 〈\latexrelease〉 \let\@input@file@exists@with@hooks\@undefined
199 〈\latexrelease〉 \let\unqu@tefilef@und\@undefined
200 〈\latexrelease〉 \EndIncludeInRelease
201 〈*2ekernel〉

```

(End of definition for \InputIfFileExists, \@input@file@exists@with@hooks, and  
\unqu@tefilef@und.)

## 2.5 Declaring a file substitution

```

202 〈@=filehook〉
203 〈/2ekernel〉
204 〈*2ekernel | \ latexrelease〉
205 〈\latexrelease〉 \IncludeInRelease{2020/10/01}%
206 〈\latexrelease〉 { __filehook_subst_add:nn } { Declaring file substitution } %
207 〈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).

```

208 \cs_new_protected:Npn __filehook_subst_add:nn #1 #2
209 {
210 \group_begin:
211 \cs_set:cpx { } { \exp_not:o { \cs:w\cs_end: } }
212 \int_set:Nn \tex_escapechar:D { -1 }
213 \cs_gset:cpx
214 {
215 @file-subst@
216 __filehook_subst_file_normalize:Nn \use_i_i_iii:nnn {#1}
217 }
218 { __filehook_subst_file_normalize:Nn __filehook_file_name_compose:nnn
219 {#2} }
220 \group_end:
221 }
222 \cs_new_protected:Npn __filehook_subst_remove:n #1
223 {

```

```

224 \group_begin:
225 \cs_set:cpx { } { \exp_not:o { \cs:w\cs_end: } }
226 \int_set:Nn \tex_escapechar:D { -1 }
227 \cs_undefine:c
228 {
229 @file-subst@
230 __filehook_subst_file_normalize:Nn \use_ii_iii:n {#1}
231 }
232 \group_end:
233 }
234 \cs_new:Npn __filehook_subst_file_normalize:Nn #1 #2
235 {
236 \exp_after:wN __filehook_subst_empty_name_chk:NN
237 \cs:w \exp_after:wN \cs_end:
238 \cs:w __filehook_file_parse_full_name:nN {#2} #1 \cs_end:
239 }
240 \cs_new:Npn __filehook_subst_empty_name_chk:NN #1 #2
241 { \if_meaning:w #1 #2 .tex \else: \token_to_str:N #2 \fi: }

(End of definition for __filehook_subst_add:nn and others.)

```

\use\_ii\_iii:n A variant of \use\_... to discard the first of three arguments.

*Todo: this should move to expl3*

```
242 \cs_gset:Npn \use_ii_iii:n {#2 #3}
```

(End of definition for \use\_ii\_iii:n.)

```

243 \ExplSyntaxOff
244 ⟨/2ekernel | latexrelease⟩
245 ⟨latexrelease⟩\EndIncludeInRelease
246 ⟨*2ekernel⟩

```

\declare@file@substitution  
\undeclare@file@substitution For two internals we provide L<sup>A</sup>T<sub>E</sub>X 2<sub>C</sub> 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`).

```

247 ⟨/2ekernel⟩
248 ⟨*2ekernel | latexrelease⟩
249 ⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
250 ⟨latexrelease⟩ {\declare@file@substitution}{File substitution}%
251 \ExplSyntaxOn
252 \cs_new_eq:NN \declare@file@substitution __filehook_subst_add:nn
253 \cs_new_eq:NN \undeclare@file@substitution __filehook_subst_remove:n
254 \ExplSyntaxOff
255 ⟨/2ekernel | latexrelease⟩
256 ⟨latexrelease⟩\EndIncludeInRelease

```

We are not fully rolling back the file substitutions in case a rollback encounters a package that contains them, but is itself not setup for rollback. So we just bypass them and hope for the best.

```

257 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
258 ⟨latexrelease⟩ {\declare@file@substitution}{File substitution}%
259 ⟨latexrelease⟩
260 ⟨latexrelease⟩\let \declare@file@substitution \gobbletwo
261 ⟨latexrelease⟩\let \undeclare@file@substitution \gobbleone
262 ⟨latexrelease⟩

```

```

263 ⟨latexrelease⟩\EndIncludeInRelease
264 ⟨*2ekernel⟩

(End of definition for \declare@file@substitution and \undeclare@file@substitution. These
functions are documented on page 1040.)
```

265 ⟨@C=⟩

## 2.6 Selecting a file (\set@curr@file)

\set@curr@file Now we hook into \set@curr@file to resolve a possible file substitution, and add \Expl@@@filehook@set@curr@file@onNN at the end, after \curr@file is set.

\curr@file A file name is built using \expandafter\string\csname<filename>\endcsname to avoid expanding utf8 active characters. The \csname expands the normalization 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.

```

266 ⟨/2ekernel⟩
267 ⟨*2ekernel | latexrelease⟩
268 ⟨latexrelease⟩\IncludeInRelease{2022/06/01}%
269 ⟨latexrelease⟩ {\set@curr@file}{Setting current file name}%
270 \def\set@curr@file{%
271 \begingroup
272 \set@curr@file@aux
273 \edef\set@curr@file@nosearch{%
274 \begingroup
275 \let\noexpand\input@path\noexpand\empty
276 \csname seq_clear:N\endcsname
277 \expandafter\noexpand\csname l_file_search_path_seq\endcsname
278 \noexpand\set@curr@file@aux
279 \def\set@curr@file@aux#1{%
280 \escapechar`m@ne
281 \let\protect\string
282 \edef~{\string~}%
283 \expandafter\def\csname\expandafter\endcsname
284 \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@CTF short-cuts and just copies \curr@file, otherwise the full normalization 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.

```
285 \@expl@@@filehook@if@no@extension@nTF{#1}%
286 { \if@tempswatru{e}{\if@tempswafalse}{}%
287 \@kernel@make@file@csname\@curr@file
288 \@expl@@@filehook@resolve@file@subst@w {#1}%
289 \@expl@@@filehook@if@file@replaced@@TF
290 {\@kernel@make@file@csname\@curr@file@reqd
291 \@expl@@@filehook@normalize@file@name@w{#1}%
292 \if@tempswa \@expl@@@filehook@drop@extension@N\@curr@file@reqd \fi}%
293 {\if@tempswa \@expl@@@filehook@drop@extension@N\@curr@file \fi
294 \global\let\@curr@file@reqd\@curr@file}%
295 \@expl@@@filehook@clear@replacement@flag@@
296 \endgroup}
297 (/2ekernel | latexrelease)
298 (latexrelease)\EndIncludeInRelease

299 (latexrelease)\IncludeInRelease{2021/06/01}%
300 (latexrelease) {\set@curr@file}{Setting current file name}%
301 (latexrelease)\def\set@curr@file#1{%
302 (latexrelease) \begingroup
303 (latexrelease) \escapechar\m@ne
304 (latexrelease) \let\protect\string
305 (latexrelease) \edef~{\string~}%
306 (latexrelease) \expandafter\def\csname\expandafter\endcsname
307 (latexrelease) \expandafter{\csname\endcsname}%
308 (latexrelease) \@expl@@@filehook@if@no@extension@nTF{#1}%
309 (latexrelease) {\if@tempswatru{e}{\if@tempswafalse}{}%
310 (latexrelease) \@kernel@make@file@csname\@curr@file
311 (latexrelease) \@expl@@@filehook@resolve@file@subst@w {#1}%
312 (latexrelease) \@expl@@@filehook@if@file@replaced@@TF
313 (latexrelease) {\@kernel@make@file@csname\@curr@file@reqd
314 (latexrelease) \@expl@@@filehook@normalize@file@name@w{#1}%
315 (latexrelease) \if@tempswa \@expl@@@filehook@drop@extension@N\@curr@file@reqd \fi}%
316 (latexrelease) {\if@tempswa \@expl@@@filehook@drop@extension@N\@curr@file \fi
317 (latexrelease) \global\let\@curr@file@reqd\@curr@file}%
318 (latexrelease) \@expl@@@filehook@clear@replacement@flag@@
319 (latexrelease) \endgroup}
320 (latexrelease)\let\set@curr@file@nosearch@\undefined
321 (latexrelease)\EndIncludeInRelease

322 (latexrelease)\IncludeInRelease{2020/10/01}%
323 (latexrelease) {\set@curr@file}{Setting current file name}%
324 (latexrelease)\def\set@curr@file#1{%
325 (latexrelease) \begingroup
326 (latexrelease) \escapechar\m@ne
327 (latexrelease) \expandafter\def\csname\expandafter\endcsname
328 (latexrelease) \expandafter{\csname\endcsname}%
329 (latexrelease) \@expl@@@filehook@if@no@extension@nTF{#1}%
330 (latexrelease) {\if@tempswatru{e}{\if@tempswafalse}{}%
331 (latexrelease) \@kernel@make@file@csname\@curr@file
332 (latexrelease) \@expl@@@filehook@resolve@file@subst@w {#1}%
333 (latexrelease) \@expl@@@filehook@if@file@replaced@@TF
334 (latexrelease) {\@kernel@make@file@csname\@curr@file@reqd
335 (latexrelease) \@expl@@@filehook@normalize@file@name@w{#1}%
```

```

336 <|latexrelease> \if@tempswa \expl@@@filehook@drop@extension@N\@curr@file@reqd \fi}%
337 <|latexrelease> {\if@tempswa \expl@@@filehook@drop@extension@N\@curr@file \fi
338 <|latexrelease> \global\let@\curr@file@reqd\curr@file}%
339 <|latexrelease> \expl@@@filehook@clear@replacement@flag@@
340 <|latexrelease> \endgroup}
341 <|latexrelease>\let\set@\curr@file@nosearch@\undefined
342 <|latexrelease>\EndIncludeInRelease

343 <|latexrelease>\IncludeInRelease{2019/10/01}%
344 <|latexrelease> {\set@\curr@file}{Setting current file name}%
345 <|latexrelease>\def\set@\curr@file#1{%
346 <|latexrelease> \begingroup
347 <|latexrelease> \escapechar\m@ne
348 <|latexrelease> \xdef@\curr@file{%
349 <|latexrelease> \expandafter\expandafter\expandafter\unquote@name
350 <|latexrelease> \expandafter\expandafter\expandafter{%
351 <|latexrelease> \expandafter\string
352 <|latexrelease> \csname@\firstofone#1\empty\endcsname}%
353 <|latexrelease> \endgroup
354 <|latexrelease>}
355 <|latexrelease>\let\set@\curr@file@nosearch@\undefined
356 <|latexrelease>\EndIncludeInRelease

357 <|latexrelease>\IncludeInRelease{0000/00/00}%
358 <|latexrelease> {\set@\curr@file}{Setting current file name}%
359 <|latexrelease>\let\set@\curr@file@\undefined
360 <|latexrelease>\let\set@\curr@file@nosearch@\undefined
361 <|latexrelease>\EndIncludeInRelease
362 <|2ekernel>

```

(End of definition for `\set@\curr@file` and others.)

*Todo: This should get internalized using `\expl@` names*

```

\@filehook@set@CurrentFile
\@kernel@make@file@csname
\@set@\curr@file@aux
363 <|2ekernel>
364 <|2ekernel | latexrelease>
365 <|latexrelease>\IncludeInRelease{2020/10/01}%
366 <|latexrelease> {\@kernel@make@file@csname}{Make file csname}%

367 \def@\kernel@make@file@csname#1#2#3{%
368 \xdef#1{\expandafter\@set@\curr@file@aux
369 \csname\expandafter#2\@firstofone#3\empty\endcsname}%

```

This auxiliary compares `\<filename>` with `\csname\endcsname` to check if the empty `.tex` file was requested.

```

370 \long\def\@set@\curr@file@aux#1{%
371 \expandafter\ifx\csname\endcsname#1%
372 .tex\else\string#1\fi}

```

Then we call `\expl@@@filehook@set@curr@file@nn` 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@replaced@tf` 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.

```

373 \def@\filehook@set@CurrentFile{%

```

```

374 \Expl@@@filehook@set@curr@file@@nNN{\curr@file}%
375 \CurrentFileUsed\CurrentFilePathUsed
376 \ifx\curr@file@reqd\curr@file
377 \let\CurrentFile\CurrentFileUsed
378 \let\CurrentFilePath\CurrentFilePathUsed
379 \else
380 \Expl@@@filehook@set@curr@file@@nNN{\curr@file@reqd}%
381 \CurrentFile\CurrentFilePath
382 \fi}
383 {/2ekernel | latexrelease}
384 {latexrelease}\EndIncludeInRelease
385 {*2ekernel}

(End of definition for \filehook@set@CurrentFile, \kernel@make@file@csname, and
\set@curr@file@aux.)

```

\@@\_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.

```

386 {/2ekernel}
387 {*2ekernel | latexrelease}
388 {latexrelease}\IncludeInRelease{2020/10/01}%
389 {latexrelease} {\@@_set_curr_file:nNN}{Set curr file}%
390 \ExplSyntaxOn
391 {@@=filehook}
392 \cs_new_protected:Npn __filehook_set_curr_file:nNN #1
393 {
394 \exp_args:Nf __filehook_file_parse_full_name:nN {#1}
395 __filehook_set_curr_file_assign:nnnNN
396 }
397 \cs_new_protected:Npn __filehook_set_curr_file_assign:nnnNN #1 #2 #3 #4 #5
398 {
399 \str_set:Nn #5 {#1}
400 \str_set:Nn #4 {#2#3}
401 }
402 \ExplSyntaxOff
403 {/2ekernel | latexrelease}
404 {latexrelease}\EndIncludeInRelease
405 {*2ekernel}

(End of definition for \@@_set_curr_file:nNN and \@@_set_curr_file_assign:nnnNN.)

```

## 2.7 Replacing a file and detecting loops

Start by sanitizing the file with `\__filehook_file_parse_full_name:nN` then do `\__filehook_file_subst_begin:nnn{\path}{\name}{\ext}`.

```

406 {/2ekernel}
407 {*2ekernel | latexrelease}
408 {latexrelease}\IncludeInRelease{2020/10/01}%
409 {latexrelease} {__filehook_resolve_file_subst:w}{Replace files detect loops}%
410 \ExplSyntaxOn

```

```

411 \cs_new:Npn __filehook_resolve_file_subst:w #1 \@nil
412 { __filehook_file_parse_full_name:nN {#1} __filehook_file_subst_begin:nnn }
413 \cs_new:Npn __filehook_normalize_file_name:w #1 \@nil
414 { __filehook_file_parse_full_name:nN {#1} __filehook_file_name_compose:nnn }
415 \cs_new:Npn __filehook_file_name_compose:nnn #1 #2 #3
416 { \tl_if_empty:nF {#1} { #1 / } #2#3 }

__filehook_file_replaced
 __filehook_if_file_replaced:TF
 __filehook_clear_replacement_flag:

__filehook_file_subst_begin:nnn

```

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 optimizing). 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.

```

417 \flag_new:n { __filehook_file_replaced }
418 \cs_new:Npn __filehook_if_file_replaced:TF #1 #2
419 { \flag_if_raised:nTF { __filehook_file_replaced } {#1} {#2} }
420 \cs_new_protected:Npn __filehook_clear_replacement_flag:
421 { \flag_clear:n { __filehook_file_replaced } }

```

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.

```

422 \cs_new:Npn __filehook_file_subst_begin:nnn #1 #2 #3
423 {
424 __filehook_file_subst_tortoise_hare:nn { #2#3 } { #2#3 }
425 { __filehook_file_name_compose:nnn {#1} {#2} {#3} }
426 }
427 \ExplSyntaxOff
428 {/2ekernel | latexrelease}
429 {latexrelease}\EndIncludeInRelease
430 {*2ekernel}

```

### 2.7.1 The Tortoise and Hare algorithm

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 substitution of that substitution: 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.

```

431 〈/2ekernel〉
432 〈*2ekernel | latexrelease〉
433 〈latexrelease〉\IncludeInRelease{2020/10/01}%
434 〈latexrelease〉 {__filehook_file_subst_tortoise_hare:nn}{Tortoise and Hare}%
435 \ExplSyntaxOn
436 \cs_new:Npn __filehook_file_subst_tortoise_hare:nn #1 #2 #3
437 {
438 \cs_if_exist:cTF { @file-subst@ #2 }
439 {
440 \flag_if_raised:nF { __filehook_file_replaced }
441 { \flag_raise:n { __filehook_file_replaced } }
442 \cs_if_exist:cTF { @file-subst@ \use:c { @file-subst@ #2 } }
443 {
444 __filehook_file_subst_loop:cc
445 { @file-subst@ #1 }
446 { @file-subst@ \use:c { @file-subst@ #2 } }
447 }
448 { \use:c { @file-subst@ #2 } }
449 }
450 { #3 }
451 }
```

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.

```

452 \cs_new:Npn __filehook_file_subst_loop:NN #1 #2
453 {
454 \token_if_eq_meaning:NNTF #1 #2
455 {
456 .tex
457 __filehook_file_subst_cycle_error:cN { @file-subst@ #1 } #1
458 }
459 { __filehook_file_subst_tortoise_hare:nn {#1} {#2} {#2} }
460 }
461 \cs_generate_variant:Nn __filehook_file_subst_loop:NN { cc }
```

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.

```

462 \cs_new:Npn __filehook_file_subst_cycle_error:NN #1 #2
463 {
464 \msg_expandable_error:nnff { latex2e } { file-cycle }
465 {#1} { \use:c { @file-subst@ #1 } }
466 \token_if_eq_meaning:NNF #1 #2
```

```

467 { __filehook_file_subst_cycle_error:cN { @file-subst@ #1 } #2 }
468 }
469 \cs_generate_variant:Nn __filehook_file_subst_cycle_error:NN { c }
And the error message:
470 \msg_new:nnn { latex2e } { file-cycle }
471 { File-loop!~#1~replaced~by~#2... }

(End of definition for __filehook_resolve_file_subst:w and others.)

472 \ExplSyntaxOff
473 </2ekernel | latexrelease>
474 <|latexrelease|\EndIncludeInRelease
475 <*2ekernel>
476 <@C=›

```

## 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 \disable@package@load defines \c@pkg-disable@{package} to expand to some code #2
\reenable@package@load instead of loading the package.
\@disable@packageload@do
477 </2ekernel>
478 <*2ekernel | latexrelease>
479 <|latexrelease|\IncludeInRelease{2020/10/01}%
480 <|latexrelease> {\c@disable@package@load}{Disable packages}%
481 \def\c@enable@package@load#1#2{%
482 \global\c@nameuse{\c@pkg-disable@#1.\c@pkgextension}{#2}}

```

Here we check if a control sequence named \c@pkg-disable@{name}.sty is defined, and if so don't use the package loading code #2, but use the replacement code stored in that control sequence, write something to the log, and then prevent \c@onefilewithoptions from sanity-checking the requested package date (the \expandafter here triggers one in \c@onefilewithoptions that ends a conditional there, and the \c@gobbletwo removes the date checking code from the input stream).

```

483 \def\c@enable@packageload@do#1#2{%
484 \c@ifundefined{\c@pkg-disable@#1}%
485 {#2}%
486 {\c@nameuse{\c@pkg-disable@#1}{%
487 \c@latex@info{Package '#1' has been disabled.%}
488 \MessageBreak Load request ignored}%
489 \expandafter\c@gobbletwo}}}
\reenable@package@load undefines \c@pkg-disable@{package} to reallow loading
a package.
490 \def\c@reenable@package@load#1{%
491 \global\expandafter\let
492 \c@csname \c@pkg-disable@#1.\c@pkgextension \endc@name \c@undefined}%
493 </2ekernel | latexrelease>
494 <|latexrelease|\EndIncludeInRelease
495 <|latexrelease|\IncludeInRelease{0000/00/00}%
496 <|latexrelease> {\c@enable@package@load}{Disable packages}%
497 <|latexrelease>

```

```

498 〈\latexrelease〉\let\disable@package@load \@undefined
499 〈\latexrelease〉\let\@disable@packageload@do\@undefined
500 〈\latexrelease〉\let\reenable@package@load \@undefined
501 〈\latexrelease〉\EndIncludeInRelease
502 〈*2ekernel〉

(End of definition for \disable@package@load, \reenable@package@load, and
 \@disable@packageload@do. These functions are documented on page 1040.)

```

## 2.9 High-level interfaces for L<sup>A</sup>T<sub>E</sub>X

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) L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  names to allow for internal commands to be used outside this module (and in parts that still use L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  syntax). We have to unset the @@ since we want double “at” sign in place of double underscores.

```

503 〈@@=〉
504 〈/2ekernel〉
505 〈*2ekernel | latexrelease〉
506 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
507 〈\latexrelease〉 {\@expl@@@filehook@if@no@extension@nTF}{2e tmp interfaces}%
508 \ExplSyntaxOn
509 \cs_new_eq:NN \@expl@@@filehook@if@no@extension@nTF
510 __filehook_if_no_extension:nTF
511 \cs_new_eq:NN \@expl@@@filehook@set@curr@file@nNN
512 __filehook_set_curr_file:nNN
513 \cs_new_eq:NN \@expl@@@filehook@resolve@file@subst@w
514 __filehook_resolve_file_subst:w
515 \cs_new_eq:NN \@expl@@@filehook@normalize@file@name@w
516 __filehook_normalize_file_name:w
517 \cs_new_eq:NN \@expl@@@filehook@if@file@replaced@TF
518 __filehook_if_file_replaced:TF
519 \cs_new_eq:NN \@expl@@@filehook@clear@replacement@flag@%
520 __filehook_clear_replacement_flag:
521 \cs_new_eq:NN \@expl@@@filehook@drop@extension@N
522 __filehook_drop_extension:N
523 \cs_new_eq:NN \@expl@@@filehook@file@push@%
524 __filehook_file_push:
525 \cs_new_eq:NN \@expl@@@filehook@file@pop@%
526 __filehook_file_pop:
527 \cs_new_eq:NN \@expl@@@filehook@file@pop@assign@nnnn
528 __filehook_file_pop_assign:nnnn
529 \ExplSyntaxOff

```

This one specifically has to be undefined because it is left over in the input stream from `\InputIfFileExists` and executed when `\textrun` is loaded. It cannot be `\let` to `\undefined` otherwise it would error as well, so it is `\let` to `\relax` to be silently ignored when loading `\textrun`.

```

530 </2ekernel | \textrun>
531 <\textrun>\EndIncludeInRelease
532 <\textrun>
533 <\textrun>\IncludeInRelease{0000/00/00}%
534 <\textrun> {\@expl@@filehook@if@no@extension@nTF}{2e tmp interfaces}%
535 <\textrun>\let\@expl@@filehook@file@pop@@\relax
536 <\textrun>\EndIncludeInRelease
537 <*2ekernel>
```

This ends the kernel code in this file.

```
538 </2ekernel>
```

### 3 A sample package for structuring the log output

```

539 <*structuredlog>
540 <@=filehook>
541 \ProvidesExplPackage
542 {structuredlog}{\ltfilehookdate}{\ltfilehookversion}
543 {Structuring the TeX transcript file}
```

`\g_filehook_nesting_level_int`

Stores the current package nesting level.

```
544 \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.

```

545 \int_gset:Nn \g_filehook_nesting_level_int
546 { (\tl_count:N \currnamestack) / 3 - 1 }
```

(End of definition for `\g_filehook_nesting_level_int`.)

`\_filehook_log_file_record:n`

This macro is responsible for increasing and decreasing 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.

```

547 \cs_new_protected:Npn _filehook_log_file_record:n #1
548 {
549 \str_if_eq:nnT {#1} {START} { \int_gincr:N \g_filehook_nesting_level_int }
550 \iow_term:x
551 {
552 \prg_replicate:nn { \g_filehook_nesting_level_int } { = } ~
553 (LEVEL ~ \int_use:N \g_filehook_nesting_level_int \c_space_tl #1) ~
554 \CurrentFileUsed
```

If there was a file replacement, show that as well:

```

555 \str_if_eq:NNF \CurrentFileUsed \CurrentFile
556 { ~ (\CurrentFile \c_space_tl requested) }
557 \iow_newline:
558 }
559 \str_if_eq:nnT {#1} {STOP} { \int_gdecr:N \g_filehook_nesting_level_int }
560 }
```

Now just hook the macro above in the generic `file/before...`

```

561 \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.
562 \AddToHookNext{file/after}
563 { \AddToHook{file/after}{ _filehook_log_file_record:n { STOP } } }
(End of definition for _filehook_log_file_record:n.)
564 <@@=
565 </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:

```

566 <*2ekernel>
567 \declare@file@substitution{atveryend.sty}{atveryend-ltx.sty}
568 </2ekernel>

```

Here is the package file we point to:

```

569 <*atveryend-ltx>
570 \ProvidesPackage{atveryend-ltx}
571 [2020/08/19 v1.0a
572 Emulation of the original atveryend package^^Jwith kernel methods]

```

Here are new definitions for its interfaces now pointing to the hooks in `\enddocument`

```

573 \newcommand\AfterLastShipout {\AddToHook{\enddocument/afterlastpage}}
574 \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`.

```

575 \newcommand\AtEndAfterFileList{\AddToHook{\enddocument/info}}
576 \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.

```

577 \ExplSyntaxOn
578 \newcommand\BeforeClearDocument[1]
579 { \AtEndDocument{#1}
580 \atveryend@DEPRECATED{BeforeClearDocument \tl_to_str:n{#1}}
581 }
582 \cs_new:Npn\atveryend@DEPRECATED #1
583 {\iow_term:x{=====~DEPRECATED~USAGE~#1~=====}}
584 \ExplSyntaxOff
(End of definition for \BeforeClearDocument.)
585 </atveryend-ltx>

```

# File 53

## ltshipout.dtx

### 1 Introduction

The code provides an interface to the `\shipout` primitive of T<sub>E</sub>X 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.<sup>53</sup>

#### 1.1 Overloading the `\shipout` primitive

---

`\shipout` With this implementation T<sub>E</sub>X’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.

Each shipout that actually happens (i.e., where the material is not discarded for one or the other reason) is recorded and the total number is available in a readonly variable and in a L<sup>A</sup>T<sub>E</sub>X counter.

---

`\RawShipout` This command implements a simplified shipout that bypasses the foreground and background hooks, e.g., only `shipout/firstpage` and `shipout/lastpage` are executed and the total shipout counters are incremented.

The command doesn’t use `\ShipoutBox` but its own private box register so that it can be used inside of shipout hooks to do some additional shipouts while already in the output routine with the current page being stored in `\ShipoutBox`. It does have access to `\ShipoutBox` if it is used in `shipout/before` (or `shipout/after`) and can use its content.

It is safe to use it in `shipout/before` or `shipout/after` but not necessarily in the other `shipout/...` hooks as they are intended for special processing.

---

<sup>53</sup>Heiko’s interfaces are emulated by the kernel code, if a document requests his packages, so older documents will continue to work.

---

`\ShipoutBox`  
`\l_shipout_box`

This box register is called `\ShipoutBox` (alternatively available via the L3 name `\l_shipout_box`).

This box is a “local” box and assignments to it should be done only locally. Global assignments (as done by some packages with older code where this is box is known as 255) may work but they are conceptually wrong and may result in errors under certain circumstances.

During the execution of `shipout/before` this box contains the accumulated material for the page, but not yet any material added by other shipout hooks. During execution of `shipout/after`, i.e., after the shipout has happened, the box also contains any background or foreground material.

Material from the hooks `shipout/firstpage` or `shipout/lastpage` is not included (but only used during the actual shipout) to facilitate reuse of the box data (e.g., `shipout/firstpage` material should never be added to a later page of the output).

---

`\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 L<sup>A</sup>T<sub>E</sub>X 2<sub><</sub> names).<sup>54</sup> These variables can be used inside the hook code for `shipout/before`, `shipout/foreground` and `shipout/background` if needed.

## 1.2 Provided hooks

---

`shipout/before`  
`shipout/after`  
`shipout/foreground`  
`shipout/background`  
`shipout/firstpage`  
`shipout/lastpage`

The code for `\shipout` 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).

You can use `\RawShipout` inside this hook for special use cases. It can make use of `\ShipoutBox` (which doesn’t yet include the background and foreground material).

**Note:** It is not possible (or say advisable) to try and use this hook to typeset material with the intention to return it to main vertical list, it will go wrong and give unexpected results in many cases—for starters it will appear after the current page not before or it will vanish or the vertical spacing will be wrong!

`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.

---

<sup>54</sup>Might need changing, but HO’s version as strings is not really helpful I think).

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** This hook is executed after foreground and/or background material has been added, i.e., just in front of the actual shipout operation. Its purpose is to allow manipulation of the finalized box (stored in `\ShipoutBox`) with the extra material also in place (which is not yet the case in `shipout/before`).

It cannot be used to cancel the shipout operation via `\DiscardShipoutBox` (that has to happen in `shipout/before`, if desired!)

**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.<sup>55</sup>

This hook is added to the very first page regardless of how it is shipped out (i.e., with `\shipout` or `\RawShipout`).

**shipout/lastpage** The corresponding hook to add `\specials` at the very end of the output file. It is only executed on the very last page of the output file — or rather on the page that `LATEX` believes is the last one. Again it is executed regardless of the shipout method.

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.

**shipout/after** This hook is executed after a shipout has happened. If the shipout box is discarded this hook is not looked at.

You can use `\RawShipout` inside this hook for special use cases and the main `\ShipoutBox` is still available at this point (but in contrast to `shipout/before` it now includes the background and foreground material).

**Note:** Just like `shipout/before` this hook is not meant to be used for adding typeset material back to the main vertical list—it might vanish or the vertical spacing will be wrong!

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. It is even run if there was a `\DiscardShipoutBox` request in the document.

The other hooks (except `shipout` and `shipout/after`) are added inside hboxes to the box being shipped out in the following order:

---

<sup>55</sup>In `LATEX 2ε` that was already existing, but implemented using a box register with the name `\@begindvbox`.

|                                                   |                        |
|---------------------------------------------------|------------------------|
| <code>shipout/firstpage</code>                    | only on the first page |
| <code>shipout/background</code>                   |                        |
| <code>&lt;boxed content of \ShipoutBox&gt;</code> |                        |
| <code>shipout/foreground</code>                   |                        |
| <code>shipout/lastpage</code>                     | only on the last page  |

If any of the hooks has no code then the corresponding box is added at that point.

Once the (page) box has got the above extra content it can again be manipulated using the `shipout` hook and then is shipped out for real.

Once the (page) box has been shipped out the `shipout/after` hook is called (while you are still inside the output routine). It is not called if the shipout box was discarded.

In a document that doesn't produce pages, e.g., only makes `\typeouts`, none of the hooks are ever executed (as there is no `\shipout`) not even the `shipout/lastpage` hook.

If `\RawShipout` is used instead of `\shipout` then only the hooks `shipout/firstpage` and `shipout/lastpage` are executed (on the first or last page), all others are bypassed.

### 1.3 Legacy L<sup>A</sup>T<sub>E</sub>X commands

---

`\AtBeginDvi` `\AtBeginDvi {<code>}`  
`\AtEndDvi`

`\AtBeginDvi` is the existing L<sup>A</sup>T<sub>E</sub>X 2<sub>E</sub> 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 `.xdv`.

`\AtEndDvi` is the counterpart that was not available in the kernel but only through the package `atenddvi`. It fills the `shipout/lastpage` hook.

Neither interface can set a code label but uses the current default label.

As these two wrappers have been available for a long time we continue offering them (but not enhancing them, e.g., by providing support for code labels).

For new code we strongly suggest using the high-level hook management commands directly instead of “randomly-named” wrappers. This will lead to code that is easier to understand and to maintain and it also allows you to set code labels if needed.

For this reason we do not provide any other “new” wrapper commands for the above hooks in the kernel, but only keep the existing ones for backward compatibility.

## 1.4 Special commands for use inside the hooks

---

```
\DiscardShipoutBox \AddToHookNext {shipout/before} {...\DiscardShipoutBox...}
```

**\shipout\_discard:**

The `\DiscardShipoutBox` declaration (L3 name `\shipout_discard:`) 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 L<sup>A</sup>T<sub>E</sub>X output routine is called in an asynchronous manner! Thus normally one would use this only as part of the `shipout/before` code.

*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 — not done (yet)*

`\DiscardShipoutBox` cannot be used in any of the `shipout/...` hooks other than `shipout/before`.

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`, `shipout/background` or `shipout/foreground`.<sup>56</sup> If `atbegshi` gets loaded then those commands become available as public functions with their original names as given below.

## 1.5 Provided LuaT<sub>E</sub>X callbacks

---

**pre\_shipout\_filter** Under LuaT<sub>E</sub>X the `pre_shipout_filter` Lua callback is provided which gets called directly after the `shipout` hook, immediately before the shipout primitive gets invoked. The signature is

```
function(<node> head)
 return true
end
```

The `head` is the list node corresponding to the box to be shipped out. The return value should always be `true`.

---

<sup>56</sup>If that assumption turns out to be wrong it would be trivial to change them to public functions (right now they are private).

## 1.6 Information counters

---

```
\ ReadonlyShipoutCounter \ifnum\ ReadonlyShipoutCounter=...
\g_shipout_READONLY_int \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)! In contrast `shipout/after` sees the incremented value.

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 restriction, but if you manipulate it, chaos will be the result. To emphasize this fact it is not provided as a L<sup>A</sup>T<sub>E</sub>X counter but as a T<sub>E</sub>X counter (i.e., a command), so `\Alph{\ReadonlyShipoutCounter}` etc, would not work.

---

```
totalpages \arabic{totalpages}
\g_shipout_totalpages_int \int_use:N \g_shipout_totalpage_int % expl3 usage
```

---

In contrast to `\ReadonlyShipoutCounter`, the `totalpages` counter is a L<sup>A</sup>T<sub>E</sub>X 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. In contrast `shipout/after` sees the incremented value.

Furthermore, while it is incremented for each page, its value is never used by L<sup>A</sup>T<sub>E</sub>X. 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 L<sup>A</sup>T<sub>E</sub>X 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 \PreviousTotalPages
```

---

Command that expands to the number of total pages from the previous run. If there was no previous run or if used in the preamble it expands to 0. Note that this is a command and not 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.

## 1.7 Debugging shipout code

---

```
\DebugShipoutsOn \DebugShipoutsOn
\DebugShipoutsOff \DebugShipoutsOff
\shipout_debug_on: \shipout_debug_on:
\shipout_debug_off: \shipout_debug_off:
```

---

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 by other packages, they are emulated when they are explicitly loaded with `\usepackage` or `\RequirePackage`.

Please note that the emulation only happens if the package is explicitly requested, i.e., the commands documented below are not automatically available in the L<sup>A</sup>T<sub>E</sub>X kernel! If you write a new package we suggest to use the appropriate kernel hooks directly instead of loading the emulation.

### 2.1 Emulating `atbegshi`

---

|                                                 |                                                                                  |
|-------------------------------------------------|----------------------------------------------------------------------------------|
| <code>\AtBeginShipoutUpperLeft</code>           | <code>\AddToHook {shipout/before} {...\AtBeginShipoutUpperLeft{\code}...}</code> |
| <code>\AtBeginShipoutUpperLeftForeground</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.

---

|                                                |                                                                                 |
|------------------------------------------------|---------------------------------------------------------------------------------|
| <code>\AtBeginShipoutAddToBox</code>           | <code>\AddToHook {shipout/before} {...\AtBeginShipoutAddToBox{\code}...}</code> |
| <code>\AtBeginShipoutAddToBoxForeground</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.

---

|                                             |
|---------------------------------------------|
| <code>\AtBeginShipoutOriginalShipout</code> |
|---------------------------------------------|

---

This is the name of the `\shipout` primitive as `atbegshi` knows it. This bypasses all the mechanisms set up by the L<sup>A</sup>T<sub>E</sub>X kernel and there are various scenarios in which it can therefore fail. It should only be used to run existing legacy `atbegshi` code but not in newly developed applications.

The kernel alternative is `\RawShipout` which is integrated with the L<sup>A</sup>T<sub>E</sub>X mechanisms and updates, for example, the `\ReadOnlyShipoutCounter` counter. Please use `\RawShipout` for new code if you want to bypass the before, foreground and background hooks.

\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 `\AtBeginShipout{\<code>} ≡ \AddToHook{shipout/before}{\<code>}`  
\AtBeginShipoutNext `\AtBeginShipoutNext{\<code>} ≡ \AddToHookNext{shipout/before}{\<code>}`

This is equivalent to filling the `shipout/before` hook by either using `\AddToHook` or `\AddToHookNext`, respectively.

\AtBeginShipoutFirst The `atbegshi` names for `\AtBeginDvi` and `\DiscardShipoutBox`.  
\AtBeginShipoutDiscard

## 2.2 Emulating `everyshi`

The `everyshi` package is providing commands to run arbitrary code just before the shipout starts. One point of difference: in the new shipout hooks the page is available as `\ShipoutBox` for inspection of change, one should not manipulate box 255 directly inside `shipout/before`, so old code doing this would change to use `\ShipoutBox` instead of 255 or `\@cclv`.

\EveryShipout `\EveryShipout{\<code>} ≡ \AddToHook{shipout/before}{\<code>}`

\AtNextShipout `\AtNextShipout{\<code>} ≡ \AddToHookNext{shipout/before}{\<code>}`

However, most use cases for `everyshi` are attempts to put some picture or text into the background or foreground of the page and that can be done today simply by using the `shipout/background` and `shipout/foreground` hooks without any need to coding.

## 2.3 Emulating `atenddvi`

The `atenddvi` package implemented only a single command: `\AtEndDvi` and that is now available out of the box so the emulation makes the package a no-op.

## 2.4 Emulating `everypage`

This package patched the original `\@begindvi` hook and replaced it with its own version. Its functionality is now covered by the hooks offered by the kernel so that there is no need for such patching any longer.

\AddEverypageHook `\AddEverypageHook{\<code>} ≡ \AddToHook{shipout/background}{\put(1in,-1in){\<code>}}`

`\AddEverypageHook` is adding something into the background of every page at a position of 1in to the right and 1in down from the top left corner of the page. By using the kernel hook directly you can put your material directly to the right place, i.e., use other coordinates in the `\put` statement above.

\AddThispageHook `\AddThispageHook{\<code>} ≡ \AddToHookNext{shipout/background}{\put(1in,-1in){\<code>}}`

The `\AddThispageHook` wrapper is similar but uses `\AddToHookNext`.

### 3 The Implementation

```
1 <@=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.

```
2 {*ekernel | latexrelease}
3 <latexrelease>\IncludeInRelease{2020/10/01}%
4 <latexrelease> {\shipout}{\Hook management (shipout)}%
5 \ExplSyntaxOn
```

#### 3.1 Debugging

\g\_\_shipout\_debug\_bool Holds the current debugging state.

```
6 \bool_new:N \g__shipout_debug_bool
```

(End of definition for \g\_\_shipout\_debug\_bool.)

\shipout\_debug\_on: Turns debugging on and off by redefining \\_\_shipout\_debug:n.

```
7 \cs_new_eq:NN __shipout_debug:n \use_none:n
8 \cs_new_protected:Npn \shipout_debug_on:
9 {
10 \bool_gset_true:N \g__shipout_debug_bool
11 __shipout_debug_gset:
12 }
13 \cs_new_protected:Npn \shipout_debug_off:
14 {
15 \bool_gset_false:N \g__shipout_debug_bool
16 __shipout_debug_gset:
17 }
18 \cs_new_protected:Npn __shipout_debug_gset:
19 {
20 \cs_gset_protected:Npx __shipout_debug:n ##1
21 { \bool_if:NT \g__shipout_debug_bool {##1} }
22 }
```

(End of definition for \shipout\_debug\_on: and others. These functions are documented on page 1064.)

\ShipoutBox The box filled with the page to be shipped out (both L3 and L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  name).

```
23 \box_new:N \l_shipout_box
24 \cs_set_eq:NN \ShipoutBox \l_shipout_box
```

(End of definition for \ShipoutBox and \l\_shipout\_box. These functions are documented on page 1060.)

\l\_\_shipout\_raw\_box The \RawShipout gets its own box but it is internal as there is no hook manipulation for it.

```
25 \box_new:N \l__shipout_raw_box
```

(End of definition for \l\_\_shipout\_raw\_box.)

\\_\\_shipout\\_finalize\\_box: For LuaTeX invoke the `pre_shipout_filter` callback.

```

26 \sys_if_engine_luatex:TF
27 {
28 \newprotectedluacmd __shipout_finalize_box:
29 \exp_args:Nx \everyjob {
30 \exp_not:V \everyjob
31 \exp_not:N \lua_now:n {
32 luatexbase.create_callback('pre_shipout_filter', 'list')
33 local~call, getbox, setbox = luatexbase.call_callback, tex.getbox, tex.setbox~
34 lua.get_functions_table() [\the \allocationnumber] = function()
35 local~head = getbox(\the \l_shipout_box)
36 local~result = call('pre_shipout_filter', head)
37 if~not (result == head) then~
38 setbox(\the \l_shipout_box, result~or~nil)
39 end~
40 end
41 }
42 }
43 } {
44 \cs_set_eq:NN __shipout_finalize_box: \scan_stop:
45 }
```

(End of definition for `\_\_shipout_finalize_box:..`)

\\_\\_shipout\\_execute: This is going to be the code run by `\shipout`. The code follows closely the ideas from atbegshi, so not documenting that here for now.

```

46 \cs_set_protected:Npn __shipout_execute: {
47 \tl_set:Nx \l__shipout_group_level_tl
48 { \int_value:w \tex_currentgrouplevel:D }
49 \tex_afterassignment:D __shipout_execute_test_level:
50 \tex_setbox:D \l_shipout_box
51 }
```

(End of definition for `\_\_shipout_execute:..`)

**\shipout** Overloading the `\shipout` primitive:

```
52 \cs_gset_eq:NN \shipout __shipout_execute:
```

(End of definition for `\shipout`. This function is documented on page 1059.)

\l\_\_shipout\_group\_level\_tl Helper token list to record the group level at which `\_\_shipout_execute:` is encountered.

```
53 \tl_new:N \l__shipout_group_level_tl
```

(End of 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`.

```

54 \cs_new:Npn __shipout_execute_test_level: {
55 \int_compare:nNnT
56 \l__shipout_group_level_tl < \tex_currentgrouplevel:D
57 \tex_aftergroup:D __shipout_execute_cont:
58 }
```

(End of definition for `\_\_shipout_execute_test_level:..`)

\\_\\_shipout\\_execute\\_cont: This does the actual shipout running several hooks as part of it. The code for them is passed as argument #2 to #4 to \\_\\_shipout\\_execute\\_main\\_cont:Nnnn; the first argument is the box to be shipped out.

```

59 \cs_new:Npn __shipout_execute_cont: {
60 __shipout_execute_main_cont:Nnnn
61 \l_shipout_box
62 { \hook_use:n {shipout/before} }
63 { \hook_if_empty:nF {shipout/foreground}
64 { __shipout_add_foreground_picture:n
65 { \hook_use:n {shipout/foreground} } } }
```

If the user hook for the background (shipout/background) has no code, there might still code in the kernel hook so we need to test for this too. We only test for the @kernel@before@shipout@background though. If the @kernel@after@shipout@background needs executing even if the user hook is empty then we can add another test (or the kernel could put something into the before hook).

```

66 \bool_lazy_and:nnF
67 { \hook_if_empty_p:n {shipout/background} }
68 { \tl_if_empty_p:N \@kernel@before@shipout@background }
69 { __shipout_add_background_picture:n
70 { \@kernel@before@shipout@background
71 \hook_use:n {shipout/background}
72 \@kernel@after@shipout@background } }
73 }
74 }
75 { \hook_use:n {shipout/after} }
76 }
```

(End of definition for \\_\\_shipout\\_execute\\_cont:.)

\\_\\_shipout\\_execute\\_main\\_cont:Nnnn 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 (unless it gets discarded during the process).

The three arguments hold hook code that is executed just before the actual shipout (#1), within the shipout adding background and foreground material (#2) and after the shipout has happened (#3). These are passed as arguments because the same code without those hooks is also used when doing a “raw” shipout implemented by \RawShipout. The only hook that is always executed is that for the very last page, i.e., shipout/lastpage.

First we quickly check if it is void (can’t happen in the standard L<sup>A</sup>T<sub>E</sub>X 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.<sup>57</sup>

```

77 \cs_new:Npn __shipout_execute_main_cont:Nnnn #1#2#3#4 {
78 \box_if_empty:NTF #1
79 { \@latex@warning@no@line{ 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.

```

81 % \bool_gset_false:N \g__shipout_discard_bool % setting this would disable
82 % % \DiscardShipoutBox on doc-level
```

---

<sup>57</sup>In that case we don’t reset the deadcycles, that would be up to the OR processing logic to do.

```

83 \cs_set_eq:NN __shipout_saved_protect: \protect
84 \set@typeset@protect

```

We also store the current shipout box dimension in registers, so that they can be used in the hook code.<sup>58</sup>

```

85 __shipout_get_box_size:N #1

```

Then we execute the `shipout/before` hook (or nothing in case of `\RawShipout`).

```

86 #2

```

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 `\g_shipout_READONLY_int` are in sync while the hook is executed (in the case that `totalpages` isn't manually altered or through discarding pages that is).

```

87 \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.

```

88 \bool_if:NTF \g__shipout_discard_bool
89 { \@latex@info@no@line{Completed~ page~ discarded}
90 \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.

```

91 \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?*

```

92 %
93 % \group_begin:
94 % \box_set_eq_drop:NN #1 #1
95 % \group_end:
96 \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.

```

97 { \box_if_empty:NTF #1
98 { \@latex@warning@no@line { Ignoring~ void~ shipout~ box.
99 \MessageBreak The~ shipout~ box~ was~ voided~ by~ hook~ code }
100 }

```

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.<sup>59</sup>

```

101 {

```

---

<sup>58</sup>This is not really necessary as the code could access them via `\box_ht:N`, etc., but it is perhaps convenient.

<sup>59</sup>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.

```

102 \int_gincr:N \g_shipout_READONLY_int
103 __shipout_DEBUG:n {
104 \typeout{Absolute~ page~ == \int_use:N \g_shipout_READONLY_int
105 \space (target:~ \@abspage@last)}
106 }

```

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.

```

107 __shipout_get_box_size:N #1

```

The first hook we run is the `shipout/firstpage` hook. This is only done once, then the `\__shipout_run_firstpage_hook:` command redefines itself to do nothing. If the hook contains `\specials` for integration at the top of the page they will be temporarily stored in a safe place and added later with `\__shipout_add_firstpage_specials::`.

```

108 __shipout_run_firstpage_hook:

```

Run the hooks for background and foreground or, if this is called by `\RawShipout`, copy the box `\l__shipout_raw_box` to `\l_shipout_box` so that `firstpage` and `lastpage` material gets added if necessary (that is always done to `\l_shipout_box`).

```

109 #3

```

We then run `\__shipout_add_firstpage_specials:` 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.

```

110 __shipout_add_firstpage_specials:

```

Then we check if we have to add the `shipout/lastpage` hook or the corresponding kernel hook because we have reached the last page. This test will be false for all but one (and hopefully the correct) page.

```

111 \int_compare:nNnT \@abspage@last = \g_shipout_READONLY_int
112 { \bool_lazy_and:nnF
113 { \hook_if_empty_p:n {shipout/lastpage} }
114 { \tl_if_empty_p:N \@kernel@after@shipout@lastpage }
115 { __shipout_DEBUG:n { \typeout{Executing~ lastpage~ hook-
116 on~ page~ \int_use:N \g_shipout_READONLY_int } }
117 __shipout_add_foreground_box:n
118 { \UseHook{shipout/lastpage}
119 \@kernel@after@shipout@lastpage }

```

We record that we have handled the `shipout/lastpage` hook but only if we really did.

```

120 \bool_gset_true:N \g__shipout_lastpage_handled_bool
121 }
122 }
123
124 \hook_use:n {shipout}
__shipout_finalize_box:

```

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.

```

125 \cs_set_eq:NN \protect \exp_not:N
126 \tex_shipout:D \box_use:N \l_shipout_box

```

The `\l_shipout_box` may contain the firstpage material if this was the very first shipout. That makes it unsuitable for reuse in another shipout, so as a safety measure the next command resets `\l_shipout_box` to its earlier state if that is necessary. On later pages this is then a no-op.

```
127 __shipout_drop_firstpage_specials:
```

The `shipout/after` hook (if in #4) needs to run with `\protected` commands again being executed, because that hook will “typeset” material added at the top of the next page.

```
128 \set@typeset@protect
129 #4
130 }
131 }
```

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).

```
132 \cs_set_eq:NN \protect __shipout_saved_protect:
133 }
134 }
```

*(End of definition for `\__shipout_execute_main_cont:Nnnn`.)*

`\__shipout_execute_raw:` `\__shipout_execute_test_level_raw:` This implements the “raw” shipout which bypasses the before, foreground, background and after hooks. It follows the same pattern than `\__shipout_execute_raw:` except that it finally calls `\__shipout_execute_main_cont:Nnnn` with three empty arguments, instead of the hook code.

```
135 \cs_set_protected:Npn __shipout_execute_raw: {
136 \tl_set:Nx \l__shipout_group_level_tl
137 { \int_value:w \tex_currentgrouplevel:D }
138 \tex_afterassignment:D __shipout_execute_test_level_raw:
139 \tex_setbox:D \l__shipout_raw_box
140 }
141 \cs_new:Npn __shipout_execute_test_level_raw: {
142 \int_compare:nNnT
143 \l__shipout_group_level_tl < \tex_currentgrouplevel:D
144 \tex_aftergroup:D __shipout_execute_nohooks_cont:
145 }
```

Well, not totally empty arguments, we add some debugging if we are actually doing a shipout.

```
146 \cs_new:Npn __shipout_execute_nohooks_cont: {
147 __shipout_execute_main_cont:Nnnn \l__shipout_raw_box
148 {} { __shipout_debug:n{ \typeout{Doing~ raw~ shipout~ ...} }
149 \box_set_eq:NN \l__shipout_box \l__shipout_raw_box } {}
150 }
```

*(End of definition for `\__shipout_execute_raw:` and `\__shipout_execute_test_level_raw:..`)*

**\RawShipout** The interface name for raw shipout.

```
151 \cs_gset_eq:NN \RawShipout __shipout_execute_raw:
```

*(End of definition for `\RawShipout`. This function is documented on page 1059.)*

`\__shipout_saved_protect:` Remember the current `\protect` state.

```
152 \cs_new_eq:NN __shipout_saved_protect: \protect
```

(End of definition for `\__shipout_saved_protect::`)

```
shipout/before
 shipout
shipout/after
shipout/foreground
shipout/background
shipout/firstpage
shipout/lastpage
```

Declaring all hooks for the shipout code.

```
153 \hook_new:n{shipout/before}
154 \hook_new:n{shipout}
155 \hook_new:n{shipout/after}
156 \hook_new:n{shipout/foreground}
157 \hook_new:n{shipout/background}
158 \hook_new:n{shipout/firstpage}
159 \hook_new:n{shipout/lastpage}
```

(End of definition for `shipout/before` and others. These functions are documented on page 1060.)

\@kernel@after@shipout@lastpage  
\@kernel@before@shipout@background  
\@kernel@after@shipout@background

And here are the internal kernel hooks going before or after the public ones where needed.

```
160 \let@\@kernel@after@shipout@lastpage\@empty
161 \let@\@kernel@before@shipout@background\@empty
162 \let@\@kernel@after@shipout@background\@empty
```

(End of definition for `\@kernel@after@shipout@lastpage`, `\@kernel@before@shipout@background`, and `\@kernel@after@shipout@background`.)

`\__shipout_run_firstpage_hook:`

There are three commands to handle the `shipout/firstpage` hook: `\__shipout_run_firstpage_hook:`, `\__shipout_add_firstpage_specials:` and `\__shipout_drop_firstpage_specials:`.

That hook is supposed to contain `\specials` and similar material to be placed at the very beginning of the output page and so it needs careful placing to avoid that anything else gets in front of it. And this means we have to wait with this until other hooks such as `shipout/background` have added their bits. It is also important that such `\specials` show up only on the very first page, so if this page gets saved before `\shipout` for later reuse, we have to make sure that they aren't in the saved version.

In addition the hook may also contain code to be executed "first", e.g., visible from code in `shipout/background` and this conflicts with adding the `\specials` late.

Therefore the processing is split into different parts: `\__shipout_run_firstpage_hook:` is done early and checks if there is any material in the hook.

```
163 \cs_new:Npn __shipout_run_firstpage_hook: {
164 \hook_if_empty:nTF {shipout/firstpage}
```

If not then we define the other two commands to do nothing.

```
165 {
166 \cs_gset_eq:NN __shipout_add_firstpage_specials: \prg_do_nothing:
167 \cs_gset_eq:NN __shipout_drop_firstpage_specials: \prg_do_nothing:
168 }
```

If there is material we execute inside a box, which means any `\special` will end up in that box and any other code is executed and can have side effects (as long as they are global).

```
169 {
170 \hbox_set:Nn \l__shipout_firstpage_box { \UseHook{shipout/firstpage} }
171 }
```

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.

```

172 \cs_gset_eq:NN __shipout_run_firstpage_hook: \prg_do_nothing:
173 \cs_gset:Npn __shipout_add_firstpage_material:Nn ##1 ##2 {
174 @latex@warning{ First~ page~ is~ already~ shipped~ out,~ ignoring
175 \MessageBreak \string##1 }
176 }
177 }
```

*(End of definition for \\_\_shipout\_run\_firstpage\_hook:.)*

\\_\_shipout\_add\_firstpage\_specials:  
\\_\_shipout\_drop\_firstpage\_specials:  
The \\_\_shipout\_add\_firstpage\_specials: then adds the \specials stored in \l\_\_shipout\_firstpage\_box to the page to be shipped out when the time is ready. Note that if there was no material in the shipout/firstpage hook then this command gets redefined to do nothing. But for most documents there is something, e.g., some PostScript header, or some meta data declaration, etc. so by default we assume there is something to do.

```
178 \cs_new:Npn __shipout_add_firstpage_specials: {
```

First we make a copy of the \l\_shipout\_box that we can restore it later on.

```
179 \box_set_eq:NN \l__shipout_raw_box \l_shipout_box
```

Adding something to the beginning means adding it to the background as that layer is done first in the output.

```
180 __shipout_add_background_box:n { \hbox_unpack_drop:N \l__shipout_firstpage_box }
```

After the actual shipout \\_\_shipout\_drop\_firstpage\_specials: is run to restore the earlier content of \l\_shipout\_box and then redefines itself again to do nothing.

As a final act we change the definition to do nothing next time.

```

181 \cs_gset_eq:NN __shipout_add_firstpage_specials: \prg_do_nothing:
182 }
```

The \\_\_shipout\_drop\_firstpage\_specials: is run after the shipout has occurred but before the shipout/afterpage hook is executed. That is the point where we have to restore the \ShipoutBox to its state without the shipout/firstpage material.

```

183 \cs_new:Npn __shipout_drop_firstpage_specials: {
184 \box_set_eq:NN \l_shipout_box \l__shipout_raw_box
```

If there was no such material then \\_\_shipout\_run\_firstpage\_hook: will have changed the definition to a no-op already. Otherwise this is what we do here.

```

185 \cs_gset_eq:NN __shipout_drop_firstpage_specials: \prg_do_nothing:
186 }
```

*(End of definition for \\_\_shipout\_add\_firstpage\_specials: and  
\\_\_shipout\_drop\_firstpage\_specials:.)*

\l\_\_shipout\_firstpage\_box The box to hold any firstpage \specials.

```
187 \box_new:N \l__shipout_firstpage_box
```

*(End of definition for \l\_\_shipout\_firstpage\_box.)*

\g\_\_shipout\_lastpage\_handled\_bool A boolean to signal if we have already handled the shipout/lastpage hook.

```
188 \bool_new:N \g__shipout_lastpage_handled_bool
```

*(End of 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 it is 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.

```
189 \cs_new:Npn _shipout_add_firstpage_material:Nn #1#2 {
190 \AddToHook{shipout/firstpage}{#2}
191 }
```

(End of 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.

```
192 \cs_new:Npn _shipout_get_box_size:N #1 {
193 \dim_set:Nn \l_shipout_box_ht_dim { \box_ht:N #1 }
194 \dim_set:Nn \l_shipout_box_dp_dim { \box_dp:N #1 }
195 \dim_set:Nn \l_shipout_box_wd_dim { \box_wd:N #1 }
196 \dim_set:Nn \l_shipout_box_ht_plus_dp_dim
197 { \l_shipout_box_ht_dim + \l_shipout_box_dp_dim }
198 }
```

(End of definition for `\_shipout_get_box_size:N`.)

`\l_shipout_box_ht_dim` And here are the variables set by `\_shipout_get_box_size:N`.

```
199 \dim_new:N \l_shipout_box_ht_dim
200 \dim_new:N \l_shipout_box_dp_dim
201 \dim_new:N \l_shipout_box_wd_dim
202 \dim_new:N \l_shipout_box_ht_plus_dp_dim
```

(End of definition for `\l_shipout_box_ht_dim` and others. These functions are documented on page 1060.)

`\g__shipout_discard_bool` Indicate whether or not the current page box should be discarded

```
203 \bool_new:N \g__shipout_discard_bool
```

(End of definition for `\g__shipout_discard_bool`.)

`\l__shipout_tmp_box` We need a box for the background and foreground material and a token register to remember badness settings as we disable them during the buildup below.  
`\l__shipout_saved_badness_tl`

```
204 \box_new:N \l__shipout_tmp_box
205 \tl_new:N \l__shipout_saved_badness_tl
```

(End of definition for `\l__shipout_tmp_box` and `\l__shipout_saved_badness_tl`.)

`\_shipout_add_background_box:n` In standard L<sup>A</sup>T<sub>E</sub>X 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.

```
206 \cs_new:Npn _shipout_add_background_box:n #1
207 { _shipout_get_box_size:N \l_shipout_box
```

But we start testing for a vertical box as that should be the normal case.

```
208 \box_if_vertical:NTF \l_shipout_box
209 {
```

Save current values of `\vfuzz` and `\vbadness` then change them to allow box manipulations without warnings.

```

210 \tl_set:Nx \l__shipout_saved_badness_tl
211 { \vfuzz=\the\vfuzz\relax
212 \vbadness=\the\vbadness\relax }
213 \vfuzz=\c_max_dim
214 \vbadness=\c_max_int

```

Then we reconstruct `\l_shipout_box` ...

```

215 \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
216 {

```

... the material in #1 is placed into a horizontal box with zero dimensions.

```

217 \hbox_set:Nn \l__shipout_tmp_box
218 { \l__shipout_saved_badness_tl #1 }
219 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
220 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
221 \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).

```

222 \skip_zero:N \baselineskip
223 \skip_zero:N \lineskip
224 \skip_zero:N \lineskiplimit
225 \box_use:N \l__shipout_tmp_box
226 \vbox_unpack:N \l_shipout_box

```

The `\kern` ensures that the box has no depth which is afterwards explicitly corrected.

```

227 \kern \c_zero_dim
228 }
229 \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
230 \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.*

```

231 \l__shipout_saved_badness_tl
232 }
233 {

```

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.

```

234 \box_if_horizontal:NT \l_shipout_box
235 {
236 \tl_set:Nx \l__shipout_saved_badness_tl
237 { \hfuzz=\the\hfuzz\relax
238 \hbadness=\the\hbadness\relax }
239 \hfuzz=\c_max_dim
240 \hbadness=\c_max_int
241 \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
242 {
243 \hbox_set:Nn \l__shipout_tmp_box
244 { \l__shipout_saved_badness_tl #1 }
245 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
246 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
247 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim

```

```

248 \box_move_up:nn
249 \l_shipout_box_ht_dim
250 { \box_use:N \l_shipout_tmp_box }
251 \hbox_unpack:N \l_shipout_box
252 }
253 \l__shipout_saved_badness_tl
254 }
255 }
256 }
```

(End of 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.

```

257 \cs_new:Npn _shipout_add_foreground_box:n #1
258 {
259 \box_if_vertical:NTF \l_shipout_box
260 {
261 \tl_set:Nx \l__shipout_saved_badness_tl
262 { \vfuzz=\the\vfuzz\relax
263 \vbadness=\the\vbadness\relax }
264 \vfuzz=\c_max_dim
265 \vbadness=\c_max_int
266 \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
267 {
268 \hbox_set:Nn \l__shipout_tmp_box
269 { \l_shipout_saved_badness_tl #1 }
270 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
271 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
272 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
273 \skip_zero:N \baselineskip
274 \skip_zero:N \lineskip
275 \skip_zero:N \lineskiplimit
276 \vbox_unpack:N \l_shipout_box
277 \kern -\l_shipout_box_ht_plus_dp_dim
278 \box_use:N \l__shipout_tmp_box
279 \kern \l_shipout_box_ht_plus_dp_dim
280 }
281 \l__shipout_saved_badness_tl
282 \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
283 \box_set_dp:Nn \l_shipout_box \l_shipout_box_dp_dim
284 }
285 {
286 \box_if_horizontal:NT \l_shipout_box
287 {
288 \tl_set:Nx \l__shipout_saved_badness_tl
289 { \hfuzz=\the\hfuzz\relax
290 \hbadness=\the\hbadness\relax }
291 \hfuzz=\c_max_dim
292 \hbadness=\c_max_int
293 \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
294 {
295 \hbox_unpack:N \l_shipout_box
296 \kern -\box_wd:N \l_shipout_box

```

```

297 \hbox_set:Nn \l__shipout_tmp_box
298 { \l__shipout_saved_badness_tl #1 }
299 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
300 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
301 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
302 \box_move_up:nn { \box_ht:N \l_shipout_box }
303 { \box_use:N \l__shipout_tmp_box }
304 \kern \box_wd:N \l_shipout_box
305 }%
306 \l__shipout_saved_badness_tl
307 }
308 }
309 }
```

(End of 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.

```

310 \cs_new:Npn __shipout_init_page_origins: {
311 \tl_const:Nx \c__shipout_horigin_tl
312 {
313 \cs_if_exist_use:NTF \pdfvariable { horigin }
314 { \cs_if_exist_use:NF \pdfhorigin { 1in } }
315 }
316 \tl_const:Nx \c__shipout_vorigin_tl
317 {
318 \cs_if_exist_use:NTF \pdfvariable { vorigin }
319 { \cs_if_exist_use:NF \pdfvorigin { 1in } }
320 }
321 }
```

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.

```

321 \cs_gset_eq:NN __shipout_init_page_origins: \prg_do_nothing:
322 }
```

(End of 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.*

```

323 \cs_new:Npn __shipout_picture_overlay:n #1 {
```

The very first time this is executed we have to initialize (and freeze) the origins.

```

324 __shipout_init_page_origins:
325 \kern -\c__shipout_horigin_tl \scan_stop:
326 \vbox_to_zero:n {
327 \kern -\c__shipout_vorigin_tl \scan_stop:
328 \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).

```

329 \hbox_set_to_wd:Nnn \l__shipout_tmp_box \c_zero_dim
330 { \ignorespaces #1 \hss }
331 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
332 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
333 \box_use:N \l__shipout_tmp_box
334 \tex_vss:D
335 }
336 }
```

(End of 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.

```

337 \cs_new:Npn __shipout_add_background_picture:n #1 {
338 __shipout_add_background_box:n { __shipout_picture_overlay:n {#1} }
339 }
```

(End of 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.

```

340 \cs_new:Npn __shipout_add_foreground_picture:n #1 {
341 __shipout_add_foreground_box:n { __shipout_picture_overlay:n {#1} }
342 }
```

(End of 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 L<sup>A</sup>T<sub>E</sub>X looks ahead and is not using the position for on the next page).

```

343 \cs_new_protected:Npn \shipout_discard: {
344 \bool_gset_true:N \g__shipout_discard_bool
345 }
```

(End of definition for `\shipout_discard:`. This function is documented on page 1063.)

### 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.

```
346 \int_new:N \g_shipout_READONLY_int
```

For L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> it is available as a command (i.e., a T<sub>E</sub>X counter only).

```
347 \cs_new_eq:NN \ ReadonlyShipoutCounter \g_shipout_READONLY_int
```

(End of definition for `\g_shipout_READONLY_int` and `\ ReadonlyShipoutCounter`. These functions are documented on page 1064.)

`\g_shipout_totalpages_int`  
`\c@totalpages`

We count every shipout attempt (even those that are discarded) in this counter. It is not used in the code but may get used in user code.

348 `\int_new:N \g_shipout_totalpages_int`

For L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  this is offered as a L<sup>A</sup>T<sub>E</sub>X counter so can be easily typeset inside the output routine to display things like “`\thepage/\thetotalpages`”, etc.

349 `\cs_new_eq:NN \c@totalpages \g_shipout_totalpages_int`  
350 `\cs_new:Npn \thetotalpages { \arabic{totalpages} }`

(End of definition for `\g_shipout_totalpages_int` and `\c@totalpages`. These functions are documented on page 1064.)

`\@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 T<sub>E</sub>X. 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.

351 `\xdef\@abspage@last{\number\maxdimen}`

(End of definition for `\@abspage@last`.)

`\enddocument`

Instead of using the hooks `enddocument` and `enddocument/afterlastpage` we add this code to private kernel hooks to be 100% sure 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 chance 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`

352 `\g@addto@macro \@kernel@after@enddocument {`  
353   `\int_compare:nNnT \@abspage@last = \maxdimen`  
354   `{`

We use L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  coding as `\@abspage@last` is not an L3 name.

355   `\xdef\@abspage@last{ \int_eval:n { \g_shipout_READONLY_int + 1 } }`  
356   `}`  
357 `}`

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.

`\@kernel@after@enddocument@afterlastpage`

358 `\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.

359   `\int_compare:nNnF \g_shipout_READONLY_int = 0`  
360   `{`

This ends up in the .aux so we use L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  names here.

*Todo: This needs an interface for \nofiles in expl3, doesn't at the moment!*

```

361 \if@filesw
362 \iow_now:Nx \@auxout {
363 \gdef\string\@abspage@last {\int_use:N \g_shipout_READONLY_int}
364 \fi

```

But we may have guessed wrongly earlier and have run it too early or 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. In either case we should put out an appropriate “rerun” warning.

```

365 \bool_if:NTF \g__shipout_lastpage_handled_bool
366 {

```

If the hook was already executed, we have to test if that total shipouts match the shipouts from last run (because that corresponds to the page it was executed). If not we output a warning.

```

367 \int_compare:nNnF \@abspage@last = \g_shipout_READONLY_int
368 {
369 \@latex@warning@no@line{Hook~ 'shipout/lastpage'~ executed-
370 on~ wrong~ page~ (\@abspage@last\space not-
371 \int_use:N\g_shipout_READONLY_int). \MessageBreak
372 Rerun~ to~ correct~ this}%
373 }
374 }
375 {

```

If the hook was not run, we need to add an extra page and place it there. 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.

```

376 \bool_lazy_and:nnF
377 { \hook_if_empty_p:n {shipout/lastpage} }
378 { \tl_if_empty_p:N \@kernel@after@shipout@lastpage }
379 {
380 \tex_shipout:D\vbox to\textheight
381 {
382 \hbox:n { \UseHook{shipout/lastpage}
383 \@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.

```

384 __shipout_excuse_extra_page:
385 \null
386 }

```

At this point we also signal to L<sup>A</sup>T<sub>E</sub>X'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 in `\enddocument`.

```

387 \cs_gset_eq:NN \@extra@page@added \relax
388 }
389 }
390 }
391 }

```

(End of definition for `\enddocument`, `\@kernel@after\enddocument`, and  
`\@kernel@after\enddocument@afterlastpage`.)

```

_shipout_excuse_extra_page: Say mea culpa ...
392 \cs_new:Npn _shipout_excuse_extra_page: {
393 \vfil
394 \begin{center}
395 \bfseries Temporary~ page!
396 \end{center}
397 \LaTeX{} was~ unable~ to~ guess~ the~ total~ number~ of~ pages~
398 correctly.~ ~ As~ there~ was~ some~ unprocessed~ data~ that~
399 should~ have~ been~ added~ to~ the~ final~ page~ this~ extra~
400 page~ has~ been~ added~ to~ receive~ it.
401 \par
402 If~ you~ rerun~ the~ document~ (without~ altering~ it)~ this~
403 surplus~ page~ will~ go~ away,~ because~ \LaTeX{} now~ knows~
404 how~ many~ pages~ to~ expect~ for~ this~ document.
405 \vfil
406 }
```

(End of definition for `\_shipout_excuse_extra_page`.)

`\PreviousTotalPages` In the preamble before the aux file was read `\PreviousTotalPages` is always zero.  
`\@kernel@before\begindocument`

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.

```

408 \g@addto@macro\@kernel@before\begindocument
409 {\ifnum\@abspage@last<\maxdimen
410 \xdef\PreviousTotalPages{\@abspage@last}\fi}
```

(End of definition for `\PreviousTotalPages` and `\@kernel@before\begindocument`. These functions are documented on page [1064](#).)

## 4 Legacy L<sup>A</sup>T<sub>E</sub>X 2<sub>&</sub> interfaces

`\DiscardShipoutBox` Request that the next shipout box is to be discarded.

```

411 \cs_new_eq:NN \DiscardShipoutBox \shipout_discard:
```

(End of definition for `\DiscardShipoutBox`. This function is documented on page [1063](#).)

`\AtBeginDvi` If we roll forward from an earlier kernel `\AtBeginDvi` is defined so we better not use `\cs_new_protected:Npn` here.

```

412 \cs_set_protected:Npn \AtBeginDvi
413 {_shipout_add_firstpage_material:Nn \AtBeginDvi}
```

(End of definition for `\AtBeginDvi`. This function is documented on page [1062](#).)

`\DebugShipoutsOn`

`\DebugShipoutsOff`

```

414 \cs_new_eq:NN \DebugShipoutsOn \shipout_debug_on:
415 \cs_new_eq:NN \DebugShipoutsOff \shipout_debug_off:
```

(End of definition for `\DebugShipoutsOn` and `\DebugShipoutsOff`. These functions are documented on page [1064](#).)

## 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.

416 <@@= >

Some internals needed elsewhere.

```
417 \cs_set_eq:NN \cexpl@@@shipout@add@firstpage@material@@Nn
418 __shipout_add_firstpage_material:Nn
419 \cs_set_eq:NN \cexpl@@@shipout@add@background@box@@n
420 __shipout_add_background_box:n
421 \cs_set_eq:NN \cexpl@@@shipout@add@foreground@box@@n
422 __shipout_add_foreground_box:n
423 \cs_set_eq:NN \cexpl@@@shipout@add@background@picture@@n
424 __shipout_add_background_picture:n
425 \cs_set_eq:NN \cexpl@@@shipout@add@foreground@picture@@n
426 __shipout_add_foreground_picture:n
```

(End of definition for \cexpl@@@shipout@add@firstpage@material@@Nn and others.)

```
427 \ExplSyntaxOff
428 {/ekernel | latexrelease}
429 {/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.

```
430 {/latexrelease}\IncludeInRelease{0000/00/00}%
431 {/latexrelease} {\shipout}{Hook management (shipout)}%
432 {/latexrelease}
```

If we roll forward then \tex\_shipout:D may not be defined in which case \shipout does have its original definition and so we must not \let it to something else which is \relax!

```
433 {/latexrelease}\ifcsname tex_shipout:D\endcsname
434 {/latexrelease}\expandafter\let\expandafter\shipout
435 {/latexrelease} \csname tex_shipout:D\endcsname
436 {/latexrelease}\fi
437 {/latexrelease}
438 {/latexrelease}\let \RawShipout\@undefined
439 {/latexrelease}\let \ShipoutBox\@undefined
440 {/latexrelease}\let \ ReadonlyShipoutCounter \@undefined
441 {/latexrelease}\let \c@totalpages \@undefined
442 {/latexrelease}\let \thetotalpages \@undefined
443 {/latexrelease}
444 {/latexrelease}\let \DiscardShipoutBox \@undefined
445 {/latexrelease}\let \DebugShipoutsOn \@undefined
446 {/latexrelease}\let \DebugShipoutsOff \@undefined
447 {/latexrelease}
448 {/latexrelease}\DeclareRobustCommand \AtBeginDvi [1]{%
449 {/latexrelease} \global \setbox \@begindvibox
450 {/latexrelease} \vbox{\unvbox \@begindvibox #1}%
451 {/latexrelease}}
```

```

452 〈\latexrelease〉
453 〈\latexrelease〉\let \AtBeginShipout \@undefined
454 〈\latexrelease〉\let \AtBeginShipoutNext \@undefined
455 〈\latexrelease〉
456 〈\latexrelease〉\let \AtBeginShipoutFirst \@undefined
457 〈\latexrelease〉
458 〈\latexrelease〉\let \ShipoutBoxHeight \@undefined
459 〈\latexrelease〉\let \ShipoutBoxDepth \@undefined
460 〈\latexrelease〉\let \ShipoutBoxWidth \@undefined
461 〈\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 it should not interfere with existing commands) and properly removing it would mean we need to make the `\undeclare@...` and its support macros available in all earlier kernel releases which is pointless (and actually worse).

```

462 〈\latexrelease〉
463 〈\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 interfere 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).

```

464 %\reenable@package@load{atenddvi}
465 〈\latexrelease〉
466 〈\latexrelease〉\EndIncludeInRelease
467 〈*2ekernel〉

```

## 6 Package emulation for compatibility

### 6.1 Package `atenddvi` emulation

`\AtEndDvi` This package has only one public command, so simulating it is easy and actually sensible to provide as part of the kernel.

```

468 〈/2ekernel〉
469 〈*2ekernel | latexrelease〉
470 〈\latexrelease〉\IncludeInRelease{2020/10/01}%
471 〈\latexrelease〉\AtEndDvi{atenddvi emulation}%
472 \ExplSyntaxOn
473 \cs_new_protected:Npn \AtEndDvi #1 {\AddToHook{shipout/lastpage}{#1}}
474 \ExplSyntaxOff

```

As the package is integrate we prevent loading (no need to roll that back):

```

475 \disable@package@load{atenddvi}
476 \PackageWarning{atenddvi}
477 {Functionality of this package is already\MessageBreak
478 provided by LaTeX.\MessageBreak\MessageBreak
479 It is there no longer necessary to load it.\MessageBreak
480 and you can safely remove it.\MessageBreak
481 Found on}%
482 〈/2ekernel | latexrelease〉

```

```

483 〈\latexrelease〉\EndIncludeInRelease
484 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
485 〈\latexrelease〉 {＼AtEndDvi}{atenddvi emulation}%
486 〈\latexrelease〉\let \AtEndDvi \undefined
487 〈\latexrelease〉\EndIncludeInRelease
488 〈*2ekernel〉

(End of definition for \AtEndDvi. This function is documented on page 1062.)

489 〈/2ekernel〉

```

## 6.2 Package **atbegshi** emulation

```

490 〈*atbegshi-ltx〉
491 〈ProvidesPackage{atbegshi-ltx}〉
492 [2021/01/10 v1.0c
493 Emulation of the original atbegshi^Jpackage with kernel methods]

```

**\AtBeginShipoutBox**

```
494 \let \AtBeginShipoutBox \ShipoutBox
```

(*End of definition for \AtBeginShipoutBox. This function is documented on page 1065.*)

**\AtBeginShipoutInit**

Compatibility only, we aren't delaying ...

```
495 \let \AtBeginShipoutInit \empty
```

(*End of definition for \AtBeginShipoutInit. This function is documented on page 1066.*)

**\AtBeginShipout**

Filling hooks

```

496 \protected\long\def\AtBeginShipout #1{\AddToHook{shipout/before}{#1}}
497 \protected\long\def\AtBeginShipoutNext #1{\AddToHookNext{shipout/before}{#1}}

```

(*End of definition for \AtBeginShipout and \AtBeginShipoutNext. These functions are documented on page 1066.*)

**\AtBeginShipoutFirst**

Slightly more complex as we need to know the name of the command under which the shipout/firstpage hook is filled.

```

498 \protected \def \AtBeginShipoutFirst
499 {\Expl@@@shipout@add@firstpage@material@Nn \AtBeginShipoutFirst}

```

(*End of definition for \AtBeginShipoutFirst. This function is documented on page 1066.*)

**\AtBeginShipoutDiscard**

Just a different name.

```
500 \let \AtBeginShipoutDiscard \DiscardShipoutBox
```

(*End of definition for \AtBeginShipoutDiscard. This function is documented on page 1066.*)

**\AtBeginShipoutAddToBox**

We don't expose them.

```

501 \let \AtBeginShipoutAddToBox
502 \Expl@@@shipout@add@background@box@Nn
503 \let \AtBeginShipoutAddToBoxForeground
504 \Expl@@@shipout@add@foreground@box@Nn
505 \let \AtBeginShipoutUpperLeft
506 \Expl@@@shipout@add@background@picture@Nn
507 \let \AtBeginShipoutUpperLeftForeground
508 \Expl@@@shipout@add@foreground@picture@Nn

```

(End of definition for \AtBeginShipoutAddToBox and others. These functions are documented on page 1065.)

\AtBeginShipoutOriginalShipout This offers the raw \shipout primitive of the engine. A page shipped out with this is not counted by \ ReadonlyShipoutCounter counter and thus the mechanism to place \specials at the very end of the output might fail, etc. It should therefore not be used in new applications but is only provided to allow running legacy code. For new code use the commands provided by the kernel instead.

```
509 \ExplSyntaxOn
510 \cs_new_eq:NN \AtBeginShipoutOriginalShipout \tex_shipout:D
```

(End of definition for \AtBeginShipoutOriginalShipout. This function is documented on page 1065.)

\ShipoutBoxHeight \ShipoutBoxWidth \ShipoutBoxDepth 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.

```
511 \cs_new:Npn \ShipoutBoxHeight { \dim_use:N \l_shipout_box_ht_dim }
512 \cs_new:Npn \ShipoutBoxDepth { \dim_use:N \l_shipout_box_dp_dim }
513 \cs_new:Npn \ShipoutBoxWidth { \dim_use:N \l_shipout_box_wd_dim }
514 \ExplSyntaxOff
```

(End of definition for \ShipoutBoxHeight, \ShipoutBoxWidth, and \ShipoutBoxDepth.)

```
515
```

If the package is requested we substitute the one above:

```
516 {*2ekernel}
517 \declare@file@substitution{atbegshi.sty}{atbegshi-ltx.sty}
518
```

### 6.3 Package **everyshi** emulation

This is now directly handled in that package so emulation is not necessary any more.

Rather important :-)

```
519
```

# File 54

## 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.

```
1 <defx>\begingroup
2 <defx>\makeatletter
3 <defx>\nfss@catcodes
4 <2ekernel>\expandafter\let\csname ver@autoout1.sty\endcsname\fmtversion
```

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

```
5 <*2ekernel>
6 \message{output,}

* OUTPUT


```

#### PAGE LAYOUT PARAMETERS

```
\topmargin : Extra space added to top of page.
@twoside : boolean. T if two-sided printing
\oddsidemargin : IF @twoside = T
 THEN extra space added to left of odd-numbered
 pages.
 ELSE extra space added to left of all pages.
\evensidemargin : IF @twoside = T
 THEN extra space added to left of even-numbered
 pages.
\headheight : height of head
\headsep : separation between head and text
\footskip : distance separation between baseline of last
 line of text and baseline of foot.
 Note difference between \footSKIP and \headSEP.
\textheight : height of text on page, excluding head and foot
\textwidth : width of printing on page
\columnsep : IF @twocolumn = T
 THEN width of space between columns
\columnseprule : IF @twocolumn = T
 THEN width of rule between columns (0 if none).
\columnwidth : IF @twocolumn = T
 THEN ($\textwidth - \columnsep$)/2
 ELSE \textwidth
 It is set by the \twocolumn and
 \onecolumn commands.
```

- \@textbottom : Command executed at bottom of vbox holding text of page (including figures). The \raggedbottom command almost \let's this to \vfil (actually sets it to \vskip \z@ plus.0001fil). Should have depth 0pt.
- \@texttop : Command executed at top of vbox holding text of page (including figures). Used by letter style; can also be used to produce centered pages. Let to \relax by \raggedbottom and \flushbottom.

Page layout must initialize \@colht and \@colroom to \textheight.

#### PAGE STYLE PARAMETERS:

- \floatsep : Space left between floats.  
 \textfloatsep : Space between last top float or first bottom float and the text.  
 \topfigrule : Command to place rule (or whatever) between floats at top of page and text. Executed in inner vertical mode right before the \textfloatsep skip separating the floats from the text. Must occupy zero vertical space. (See \footnoterule.)  
 \botfigrule : Same as \topfigrule, but put after the \textfloatsep skip separating text from the floats at bottom of page.  
 \intextsep : Space left on top and bottom of an in-text float.  
 \dblfloatsep : Space between double-column floats.  
 \dbltextfloatsep : Space between top double-column floats and text.  
 \dblfigrule : Similar to \topfigrule, but for double-column floats.  
 \@fptop : Glue to go at top of float column – must be 0pt + stretch  
 \@fpsep : Glue to go between floats in a float column.  
 \@fpbot : Glue to go at bottom of float column – must be 0pt + stretch  
 \@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 `\@dblftop`

`\@fpsep` :=L `\@dblfpsep`

`\@fpbot` :=L `\@dblfpbot`

## OUTPUT ROUTINE VARIABLES

`\@colht` : The total height of the current column. In single column style, it equals `\textheight`. In two-column style, it is `\textheight` minus the height of the double-column floats on the current page. MUST BE INITIALIZED TO `\textheight`.

`\@colroom` : The height available in the current column for text and footnotes. It equals `\@colht` minus the height of all floats committed to the top and bottom of the current column.

`\@textfloatsheight` : The total height of in-text floats on the current page.

`\footins` : Footnote insertion number.

`\@maxdepth` : Saved value of TeX's `\maxdepth`. Must be set when any routine sets `\maxdepth`.

## CALLING THE OUTPUT ROUTINE

---

The output routine is called either by TeX's normal page-breaking mechanism, or by a macro putting a penalty < or = -10000 in the output list. In the latter case, the penalty indicates why the output routine was called, using the following code.

| penalty | reason                                                                                          |
|---------|-------------------------------------------------------------------------------------------------|
| -10000  | <code>\pagebreak</code><br><code>\newpage</code>                                                |
| -10001  | <code>\clearpage</code> ( <code>\penalty -10000 \vbox{}</code> ) <code>\penalty -10001</code> ) |
| -10002  | float insertion, called from horizontal mode                                                    |
| -10003  | float insertion, called from vertical mode.                                                     |
| -10004  | float insertion.                                                                                |

Note: A float or marginpar puts the following sequence in the output list:

- (i) a penalty of -10004,
- (ii) a null \vbox
- (iii) a penalty of -10002 or -10003.

This solves two special problems:

1. If the float comes right after a \newpage or \clearpage, then the first penalty is ignored, but the second one invokes the output routine.
2. If there is a split footnote on the page, the second 'page' puts out the rest of the footnote.

## THE OUTPUT ROUTINE

---

### FUNCTIONS USED IN THE OUTPUT ROUTINE:

\@outputpage : Produces an output page with the contents of box \@outputbox as the text part.

Also sets \@colht :=G \textheight.

The page style is determined as follows.

IF @thispagestyle = true  
THEN use \thispagestyle style  
ELSE use ordinary page style.

\@tryfcolumn\FLIST : Tries to form a float column composed of floats from \FLIST (if nonempty) with the following parameters:

\@colht : height of box  
\@fpmin : minimum height of floats in the box  
\@fpsep : interfloat space  
\@fptop : glue at top of box  
\@fpbot : glue at bottom of box.

If it succeeds, then it does the following:

\* \@outputbox :=L the composed float box.  
\* @fcollmade :=G true  
\* \FLIST :=G \FLIST - floats put in box  
\* \@freelist :=G \@freelist + floats put in box

If it fails, then:

\* @fcollmade :=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) @fcollmade = 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 `\@dbltopl` 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 `\@topl` and `\@botl`, 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 `\@topl` and `\@botl`, 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 `\@dbltopl` 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  $\log_2 \NUM$  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  $\log_2 \NUM$  of its float specifier equal to 1.

Note:  $\log_2 [(\count I)/32]$  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.

`\@addtobdblcol` : Tries to add `\@currbox` to the next double-column page,  
  adding it to `\@dbltoplist` if it succeeds and  
  `\@dbldefeolist` 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
 + \marginpush
```

```

\@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 @tempcpta > 0 then \hskip \columnwidth
 \hskip \marginparsep
 else \hskip -\marginparsep
 \hskip -\marginparwidth
 fi
 \box\@marbox \hss
}
\nobreak
\nointerlineskip
\hbox{\vrule height 0 width 0 depth \@pagedp}
END

```

FLOATS AND MARGINPARS ADD A LOT OF DEAD CYCLES.

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

7 \maxdeadcycles = 100
8 \let\@elt\relax
9 \def\@next#1#2#3#4{\ifx#2\empty #4\else
10 \expandafter\@next #2\@#1#2#3\fi}
11 \def\@xnext \@elt #1#2\@#3#4{\def#3{#1}\gdef#4{#2}}
12 \def\@testfalse{\global\let\if@test\iffalse}
13 \def\@testtrue {\global\let\if@test\iftrue}
14 \qquad\@testfalse
15 \def\@bitor#1#2{\@testfalse {\let\@elt\@xbitor
16 \@tempcpta #1\relax #2}}

```

RmS 91/11/22: Added test for \count#1 = 0. Suggested by Chris Rowley.

```

17 \def\@xbitor #1{\@tempcntb \count#1
18 \ifnum \@tempcpta =\z@
19 \else
20 \divide\@tempcntb\@tempcpta
21 \ifodd\@tempcntb \@testtrue\fi
22 \fi}

```

#### DEFINITION OF FLOAT BOXES:

```

23 </2ekernel>
24 <|latexrelease>\IncludeInRelease{2015/10/01}%
25 <|latexrelease> \bx@ZZ-{Extended float list}%
26 <*2ekernel | latexrelease>
27 \let\@elt\newinsert
28 <*2ekernel>
29 \def\@freelist{%

```

```

30 \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
31 \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
32 \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
33 \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
34 \@freelist
35 </2ekernel>
36 \ifx\numexpr\@undefined\else
37 \def\reserved@a{%
38 \@elt\bx@S\@elt\bx@T\@elt\bx@U\@elt\bx@V
39 \@elt\bx@W\@elt\bx@X\@elt\bx@Y\@elt\bx@Z
40 \@elt\bx@AA\@elt\bx@BB\@elt\bx@CC\@elt\bx@DD\@elt\bx@EE
41 \@elt\bx@FF\@elt\bx@GG\@elt\bx@HH\@elt\bx@II\@elt\bx@JJ
42 \@elt\bx@KK\@elt\bx@LL\@elt\bx@MM\@elt\bx@NN
43 \@elt\bx@OO\@elt\bx@PP\@elt\bx@QQ\@elt\bx@RR
44 \@elt\bx@SS\@elt\bx@TT\@elt\bx@UU\@elt\bx@VV
45 \@elt\bx@WW\@elt\bx@XX\@elt\bx@YY\@elt\bx@ZZ}
46 \reserved@a
47 \def\@elt{\noexpand\@elt\noexpand}
48 \edef\@freelist{\@freelist\reserved@a}
49 \fi
50 \let\reserved@a\relax
51 \let\@elt\relax
52 </2ekernel | latexrelease>
53 <latexrelease>\EndIncludeInRelease
54 <latexrelease>\IncludeInRelease{0000/00/00}%
55 <latexrelease> {\bx@ZZ}{Extended float list}%
56 <latexrelease>\def\@freelist{%
57 <latexrelease> \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
58 <latexrelease> \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
59 <latexrelease> \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
60 <latexrelease> \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
61 <latexrelease> \insc@unt=234
62 <latexrelease>\EndIncludeInRelease
63 <*2ekernel>

64 \gdef\@toplist{}
65 \gdef\@botlist{}
66 \gdef\@midlist{}
67 \gdef\@currlist{}
68 \gdef\@deferlist{}
69 \gdef\@dbltoplist{}

```

The new algorithm stores page wide floats together with column floats in a single `\@deferlist` list. We keep `\@dbldeferlist` initialised as empty so that packages that are testing for deferred floats can use the same code for old or new float handling.

```

70 \gdef\@dbldeferlist{}
 PAGE LAYOUT PARAMETERS
71 \newdimen\topmargin
72 \newdimen\oddsidemargin
73 \newdimen\evensidemargin
74 \let\@themargin=\oddsidemargin
75 \newdimen\headheight
76 \newdimen\headsep
77 \newdimen\footskip

```

```

78 \newdimen\textheight
79 \newdimen\textwidth
80 \newdimen\columnwidth
81 \newdimen\columnsep
82 \newdimen\columnseprule
83 \newdimen\marginparwidth
84 \newdimen\marginparsep
85 \newdimen\marginparpush

```

**\AtBeginDvi** We use a box register in which to put stuff that must appear before anything else 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).

```

86 \newbox\@begindvibox
87 %\DeclareRobustCommand \AtBeginDvi [1]{%
88 % \global \setbox \@begindvibox
89 % \vbox{\unvbox \@begindvibox #1}%
90 %}

```

(End of definition for `\AtBeginDvi` and `\@begindvibox`. These functions are documented on page 1062.)

**\@maxdepth** This is not the right place to set this; it needs to be set in a class/style file when `\maxdepth` is set.

Also, many settings to `\maxdepth` should be to `\@maxdepth`, probably?

```

91 \newdimen\@maxdepth
92 \@maxdepth = \maxdepth

```

(End of definition for `\@maxdepth`.)

**\paperheight** New `\paper...` registers.

```

\paperwidth 93 \newdimen\paperheight
94 \newdimen\paperwidth

```

(End of definition for `\paperheight` and `\paperwidth`.)

**\stockheight** New `\stock...` registers.

```

\stockwidth 95 \newdimen\stockheight
96 \newdimen\stockwidth

```

(End of definition for `\stockheight` and `\stockwidth`.)

**\if@insert** Local switches first:

```

\if@fcolmade 97 \newif \if@insert

```

**\if@specialpage** These should definitely be global:

```

\if@firstcolumn 98 \newif \if@fcolmade
\if@twocolumn 99 \newif \if@specialpage \@specialpagefalse
\if@twoside

```

These should be global but are not always set globally in other files.

```

\if@reversemargin 100 \newif \if@firstcolumn \@firstcolumntrue
\if@mparswitch 101 \newif \if@twocolumn \@twocolumnfalse
\col@number

```

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?

```
102 \newif \if@twoside \@twosidefalse
103 \newif \if@reversemargin \@reversemarginfalse
104 \newif \if@mparswitch \@mparswitchfalse
```

This counter has been imported from ‘multicol’.

```
105 \newcount \col@number
106 \col@number \one
```

*(End of definition for \if@insert and others.)*

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

#### INTERNAL REGISTERS

```
107 \newcount\@topnum
108 \newdimen\@toproom
109 \newcount\@dbltopnum
110 \newdimen\@dbltoproom
111 \newcount\@botnum
112 \newdimen\@botroom
113 \newcount\@colnum
114 \newdimen\@textmin
115 \newdimen\@fpmin
116 \newdimen\@colht
117 \newdimen\@colroom
118 \newdimen\@pageht
119 \newdimen\@pagedp
120 \newdimen\@mparbottom \@mparbottom\z@
121 \newcount\@currtype
122 \newbox\@outputbox
123 \newbox\@leftcolumn
124 \newbox\@holdpg
```

```
125 \def\@thehead{\@oddhead} % initialization
126 \def\@thefoot{\@oddfoot}
```

*End of historical L<sup>A</sup>T<sub>E</sub>X 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.

```
127 \def\clearpage{%
128 \ifvmode
129 \ifnum \@dbltopnum =\m@ne
130 \ifdim \pagetotal <\topskip
131 \hbox{}%
132 \fi
133 \fi
134 \fi
135 \newpage
136 \write\m@ne{}}%
```

```

137 \vbox{}%
138 \penalty -\@Mi
139 }

(End of definition for \clearpage.)
```

\cleardoublepage

```

140 \def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
141 \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
142 \relax
```

(End of definition for \cleardoublepage.)

\onecolumn

```

143 \if@twokernelfltrace
144 \def\onecolumn{%
145 \clearpage
146 \global\columnwidth\textwidth
147 \global\hsize\columnwidth
148 \global\linewidth\columnwidth
149 \global\twocolumnfalse
150 \col@number @ne
151 @floatplacement}
```

(End of definition for \onecolumn.)

**\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.

```

152 \relax
153 \if@twokernelfltrace
154 \if@twokernelfltrace
155 \if@twokernelfltrace
156 \def\newpage{%
157 \if@noskipsec
158 \ifx\@nodocument\relax
159 \leavevmode
160 \global\@noskipsecfalse
161 \fi
162 \fi
163 \if@inlabel
164 \leavevmode
165 \global\@inlabelfalse
166 \fi
167 \if@nobreak\@nobreakfalse\everypar{}\fi
168 \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`.

```

169 \ifdim\prevdepth>\z@%
170 \vskip -%
171 \ifdim\prevdepth>\maxdepth
172 \maxdepth
173 \else
174 \prevdepth
175 \fi
176 \fi
177 \vfil
178 \penalty -\@M}
179 </2ekernel | latexrelease | fltrace>
180 <latexrelease>\EndIncludeInRelease
181 <latexrelease>\IncludeInRelease{0000/00/00}%
182 <latexrelease> {\newpage}{Check depth of page}%
183 <latexrelease>\def \newpage {%
184 <latexrelease> \if@noskipsec
185 <latexrelease> \ifx \@nodocument\relax
186 <latexrelease> \leavevmode
187 <latexrelease> \global \c@noskipsecfalse
188 <latexrelease> \fi
189 <latexrelease> \fi
190 <latexrelease> \if@inlabel
191 <latexrelease> \leavevmode
192 <latexrelease> \global \c@inlabelfalse
193 <latexrelease> \fi
194 <latexrelease> \if@nobreak \c@nobreakfalse \everypar{}\fi
195 <latexrelease> \par
196 <latexrelease> \vfil
197 <latexrelease> \penalty -\@M}
198 <latexrelease>\EndIncludeInRelease
199 <2ekernel | fltrace>

```

*(End of definition for `\newpage`.)*

`\@emptycol` It may be better to use an invisible rule rather than an empty box here.

```
200 \def \@emptycol {\vbox{} \penalty -\@M}
```

*(End of definition for `\@emptycol`.)*

`\twocolumn` There are several bug fixes to the two-column stuff here.

```

\@topnewpage
201 \def \twocolumn {%
202 \clearpage
203 \global \columnwidth \textwidth
204 \global \advance \columnwidth -\columnsep
205 \global \divide \columnwidth \tw@
206 \global \hsize \columnwidth
207 \global \linewidth \columnwidth
208 \global \c@twocolumntrue
209 \global \c@firstcolumntrue
210 \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.

```
211 \@ifnextchar [\@topnewpage\@floatplacement
212 }
```

Note that here, getting a box from the freelist can assume success since this comes just after a `\clearpage`.

```
213 \long\def \@topnewpage [#1]{%
214 \@nodocument
215 \@next\currbox\@freelist{}{}%
216 \global \setbox\currbox
217 \color@vbox
218 \normalcolor
219 \vbox{%
220 \hsize\textwidth
221 \parboxrestore
222 \col@number \one
223 #1%
224 \vskip -\dbltextfloatsep
225 }%
226 \color@endbox
```

Added size test and warning message; perhaps we should use an error message.

```
227 \ifdim \ht@\currbox>\textheight
228 \ht@\currbox \textheight
229 \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 `\@addtobblcol`, plus some extra checks for error trapping.

```
230 \global \count@\currbox \tw@
231 \@tempdima -\ht@\currbox
232 \advance \atempdima -\dbltextfloatsep
233 \global \advance \colht \atempdima
234 \ifx \dbltoplist \empty
235 \else
236 \@latex@error{Float(s) lost}\ehb
237 \let \dbltoplist \empty
238 \fi
239 \cons \dbltoplist \currbox
```

This setting of `\@dbltopnum` is used only to change the typesetting in `\@combinedblfloats`.

```
240 \global \dbltopnum \m@ne
241 <*trace>
242 \f@trace{dbltopnum set to -1 (= \the \dbltopnum) (topnewpage)}%
243 </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.

```

244 \ifdim \@colht<2.5\baselineskip
245 \@latex@warning@no@line {Optional argument of \noexpand\twocolumn
246 too tall on page \thepage}%
247 \@emptycol
248 \if@firstcolumn
249 \else
250 \@emptycol
251 \fi
252 \else
253 \global \vsize \@colht
254 \global \colroom \@colht
255 \@floatplacement
256 \fi
257 }
```

(End of definition for `\twocolumn` and `\@topnewpage`.)

`\output` This needs some small adjustments. We cannot guarantee that the float mechanism will interact correctly with this stuff, but that mechanism does not always work properly with footnotes already.

RmS 91/09/29:

added reset of `\par` to the output routine. This avoids problems when the output routine is called within a list where `\par` may be a no-op.

```

258 \output {%
259 \let \par \@@par
260 \ifnum \outputpenalty<-`@M
261 \@specialoutput
262 \else
263 \@makecol
264 \@opcol
```

Moved to `\@opcol`: `\@floatplacement`.

```
265 \@startcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startcolumn`.

```

266 \@whilesw \if@fcolmade \fi
267 {%
268 (*trace)
269 \f1@trace{PAGE: float \if@twocolumn column \else page \fi
270 completed}%
271 (/trace)
272 \@opcol\@startcolumn}%
273 \fi
274 \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.

```

275 \ifdim \@colroom<1.5\baselineskip
276 \ifdim \@colroom<\textheight
277 \@latex@warning@no@line {Text page \thepage\space
278 contains only floats}%
279 \@emptycol
280 %
281 \if@twocolumn
282 %
283 \else
284 %
285 \fi
286 %
287 \global \vsize \@colroom
288 %
289 \else
290 \global \vsize \@colroom
291 %
292 \else
293 \global \vsize \maxdimen
294 %
295 }
```

*Historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments (not necessarily accurate any more):*

CHANGES TO \@specialoutput:

\* \penalty\z@ changed to \penalty\interlinepenalty so \samepage works properly with figure and table environments.

(Changed 23 Oct 86)

\* Definition of \@specialoutput changed 26 Feb 88 so \@pageht and \@pagedp aren't changed for a marginal note.

(Change suggested by Chris Rowley.)

*End of historical L<sup>A</sup>T<sub>E</sub>X 2.09 comments.*

```

296 \gdef\@specialoutput{%
297 \ifnum \outputpenalty>-\@Mii
298 \@doclearpage
```

```

299 \else
300 \ifnum \outputpenalty<-\@Mii
301 \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
302 \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
303 \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.

```

304 \global \setbox\@holdpg \vbox{%
305 \unvbox\@holdpg
306 \unvbox\@cclv

```

We must now remove the box added by the float mechanism and the `\topskip` glue therefore added above it by TeX.

```

307 \setbox\@tempboxa \lastbox
308 \unskip
309 }%

```

These two are needed as separate dimensions only by `\@addmarginpar`; for other purposes we put the whole size into `\@pageht` (see below).

```

310 \@pagedp \dp\@holdpg
311 \@pageht \ht\@holdpg
312 \unvbox \@holdpg
313 \next\currbox\@currlist{%
314 \ifnum \count\currbox>\z@

```

Putting the whole size into `\@pageht` (see above).

```

315 \advance \@pageht \@pagedp
316 \ifvoid\footins \else
317 \advance \@pageht \ht\footins
318 \advance \@pageht \skip\footins
319 \advance \@pageht \dp\footins
320 \fi
321 \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?

```

322 \ifdim \wd\@kludgeins=\z@
323 \advance \@pageht \ht\@kludgeins
324 (*trace)
325 \f@trace {Extra size added: \the \ht\@kludgeins}%
326 (/trace)
327 \fi
328 \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.

```

329 \@reinserts
330 \@addtocurcol
331 \else
332 \@reinserts
333 \@addmarginpar
334 \fi
335 }\@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.

```

336 \ifnum \outputpenalty<\z@
337 \if@nobreak
338 \nobreak
339 \else
340 \addpenalty \interlinepenalty
341 \fi
342 \fi
343 \fi
344 \fi
345 }
346 </2ekernel | fltrace>

```

(End of definition for `\output` and `\@specialoutput`.)

`\@testwrongwidth` 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).

```

347 <latexrelease>\IncludeInRelease{2015/01/01}%
348 <latexrelease> {\@testwrongwidth}{float order in 2-column}%
349 <*2ekernel | latexrelease | fltrace>

350 \def\@testwrongwidth #1{%
351 \ifdim\dp#1=\f@depth
352 {*trace}
353 \f1@trace{\string#1
354 \ifdim\f@depth=\z@ single \else double \fi
355 column float -- ok}%
356 /trace}
357 \else
358 \global\@testtrue
359 {*trace}
360 \f1@trace{\string#1
361 \ifdim\f@depth=\z@ double \else single \fi
362 column float -- wrong}%
363 /trace}
364 \fi}%

```

Normally looking for single column floats, which have zero depth.  
`\let\f@depth\z@`

```

366 {/2ekernel | latexrelease | fltrace}
367 \end{IncludeInRelease}
368 \IncludeInRelease{0000/00/00}%
369 \begin{IncludeInRelease}
370 {\@testwrongwidth}{float order in 2-column}%
371 \let\@testwrongwidth\undefined
372 \let\f@depth\undefined
373 \end{IncludeInRelease}

```

(End of definition for `\@testwrongwidth` and `\f@depth`.)

### \@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.

```

373 \IncludeInRelease{2015/01/01}{\@doclearpage}%
374 \begin{IncludeInRelease}
375 {*2ekernel | latexrelease}
376 \def \@doclearpage {%
377 \ifvoid\footins
378 \ifvbox\@kludgeins
379 {\setbox \tempboxa \box \kludgeins}%
380 \else
381 \f@trace {kludgeins box made void}%
382 \fi
383 \setbox\tempboxa\vsplit\cclv to\z@\unvbox\tempboxa
384 \setbox\tempboxa\box\cclv
385 \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
386 \global \let \@toplist \empty
387 \global \let \@botlist \empty
388 \global \colroom \colht
389 \ifx \currlist\empty
390 \else
391 \@latex@error{Float(s) lost}\ehb
392 \global \let \currlist \empty
393 \fi
394 \makefcolumn\@deferlist
395 \whilesw\if@fcollmade \fi{\opcol\makefcolumn\@deferlist}%
396 \if@twocolumn
397 \if@firstcolumn
398 \xdef\@deferlist{\dbltoplist\@deferlist}%
399 \global \let \dbltoplist \empty
400 \global \colht \textheight
401 \begingroup
402 \dblfloplacement

```

```

404 \@makefcolumn\@deferlist
405 \@whilesw\if@fcolmade \fi{\@outputpage
406 \@makefcolumn\@deferlist}%
407 \endgroup
408 \else
409 \vbox{}\clearpage
410 \fi
411 \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.

```

412 \ifx\@deferlist\@empty \else\clearpage \fi
413 \else
414 \setbox\@cclv\vbox{\box\@cclv\vfil}%
415 \@makecol\@opcol
416 \clearpage
417 \fi
418 }%
419 </2ekernel | latexrelease>
420 <latexrelease>\EndIncludeInRelease
421 <latexrelease>\IncludeInRelease{0000/00/00}{\@doclearpage}%
422 <latexrelease> {float order in 2-column}%
423 <latexrelease>\def \@doclearpage {%
424 <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.

```

425 <latexrelease> \ifvbox\@kludgeins
426 <latexrelease> {\setbox \@tempboxa \box \@kludgeins}%
427 <*trace>
428 <latexrelease> \f1@trace {\kludgeins box made void}%
429 </trace>
430 <latexrelease> \fi
431 <latexrelease> \setbox\@tempboxa\vsplit\@cclv to\z@\unvbox\@tempboxa
432 <latexrelease> \setbox\@tempboxa\box\@cclv
433 <latexrelease> \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
434 <latexrelease> \global \let \@toplist \@empty
435 <latexrelease> \global \let \@botlist \@empty
436 <latexrelease> \global \@colroom \@colht
437 <latexrelease> \ifx \@currlist\@empty
438 <latexrelease> \else
439 <latexrelease> \@latexerr{Float(s) lost}\@ehb
440 <latexrelease> \global \let \@currlist \@empty
441 <latexrelease> \fi
442 <latexrelease> \@makefcolumn\@deferlist
443 <latexrelease> \@whilesw\if@fcolmade \fi
444 <latexrelease> {\@opcol\@makefcolumn\@deferlist}%
445 <latexrelease> \if@twocolumn
446 <latexrelease> \if@firstcolumn
447 <latexrelease> \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%

```

```

448 <|latexrelease> \global \let \@dbltoplist \empty
449 <|latexrelease> \global \@colht \textheight
450 <|latexrelease> \begingroup
451 <|latexrelease> \@dblfloatingplacement
452 <|latexrelease> \@makefcolumn\@dbldeflist
453 <|latexrelease> \@whilesw\if@fcolmade \fi
454 <|latexrelease> {\@outputpage\@makefcolumn\@dbldeflist}%
455 <|latexrelease> \endgroup
456 <|latexrelease> \else
457 <|latexrelease> \vbox{}\clearpage
458 <|latexrelease> \fi
459 <|latexrelease> \fi
460 <|latexrelease> \else
461 <|latexrelease> \setbox\@cclv\vbox{\box\@cclv\vfil}%
462 <|latexrelease> \@makecol\@opcol
463 <|latexrelease> \clearpage
464 <|latexrelease> \fi
465 <|latexrelease> }%
466 <|latexrelease>\EndIncludeInRelease

```

(End of definition for \@doclearpage.)

\@opcol Several changes in detail here.

```

467 <|*2ekernel | fltrace>
468 \def \@opcol {%
469 \if@twocolumn

```

The funny-looking internal commands are interfacing with the new marks mechanism. We make sure (elsewhere) that those are always defined, even when we roll back, so here we add them unconditionally. This still need turning into a hook or config point eventually:

```

470 \@expl@@@mark@update@dblcol@structures@@
471 \@outputdblcol
472 \else
473 \@expl@@@mark@update@singlcol@structures@@
474 \@outputpage
475 <|*trace>
476 \fl@trace{PAGE: one column (float? see above) page completed}%
477 </trace>

```

Not needed since it comes after \@outputpage:

```

478 % \global\@colht\textheight
479 \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.

```

480 \global \z@ \global \textfloatsheight \z@
481 \@floatplacement
482 }
483 <|/2ekernel | fltrace>

```

(End of definition for \@opcol.)

\@makecol We must rewrite this macro to allow for variations in page-makeup required by changes in page-length.

This uses a different macro if a special-length column is being produced.

```
484 {*2ekernel}
485 \gdef \@makecol {%
486 \ifvoid\footins
487 \setbox\@outputbox \box\@cclv
488 \else
489 \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.

```
490 \boxmaxdepth \@maxdepth
491 %
492 \unvbox \@cclv
493 %
494 \vskip-\@tempdima
 \vskip \skip\footins
495 \color@begingroup
 \normalcolor
 \footnoterule
 \unvbox \footins
496 \color@endgroup
497 }%
498 \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.

```
502 \let\@elt\relax
503 \xdef\@freelist{\@freelist\@midlist}%
504 \global \let \@midlist \empty
505 \combinefloats
```

The variations start here in case \enlargethispage has been used.

```
506 \ifvbox\@kludgeins
507 \makespecialcolbox
508 \else
```

This extra reboxing is only needed to add the \texttop and \textbottom but this could be done earlier, when the floats are added.

The \boxmaxdepth resetting here will have no effect unless \textbottom ends with a box or rule. So is this (or possibly \maxdepth) the correct value?

The \vskip -\dimen@ ensures that the visible depth of the box does not affect the placement of anything on the page. Thus very deep pages will overprint the footer; but these should have been prevented by suitable settings of the maxdepths at appropriate times.

If \textbottom ends with a box or rule of non-zero depth then this skip adjustment should be done again after it.

I think that the final boxing of the main text page could have a common ending which may make it simpler to see what is going on.

This needs further investigation, especially in the ‘special case’.

Also, the `\boxmaxdepth` setting here affects what happens within `\@texttop` and `\@textbottom`, should it? Is it needed at all?

RmS 91/10/22: Replaced `\dimen128` by `\dimen0`.

```
509 \setbox\@outputbox \vbox to\@colht {%
510 % \boxmaxdepth \maxdepth %??
511 \@texttop
512 \dimen0 \dp\@outputbox
513 \unvbox\@outputbox
514 \vskip -\dimen0
515 \@textbottom
516 }%
517 \fi
518 \global \maxdepth \@maxdepth
519 }
```

(End of definition for `\@makecol`.)

`\@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.

```
520 \gdef \@reinserts{%
521 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
522 \ifvbox\@kludgeins\insert\@kludgeins
523 {\unvbox\@kludgeins}\fi
524 }
525
```

(End of definition for `\@reinserts`.)

`\@makespecialcolbox` This implements certain variations in page-makeup.

```
526 (*2ekernel | fltrace)
527 \gdef \@makespecialcolbox {%
528 <*trace>
529 \fl@trace{Kludgeins ht \the\ht\@kludgeins\space
530 dp \the\dp\@kludgeins\space
531 wd \the\wd\@kludgeins}%
532 }</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’.

```
533 \setbox\@outputbox \vbox {%
534 \@texttop
535 \dimen0 \dp\@outputbox
536 \unvbox\@outputbox
537 \vskip-\dimen0
538 }%
539 \tempdima \@colht
540 \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 all the material collected before the page-break decision is made. It will, we think, always be an overestimate of the actual shrink in the box; therefore this should always force the shortest possible column with the possibility of an overfull box.

This should work for both flush- and ragged-bottom setting since it makes the contents no smaller than the size (`\@colht`) of the box into which they are put.

There 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.

```

541 \advance \@tempdima -\ht\@outputbox
542 \advance \@tempdima \pageshrink
543 {*trace}
544 \f@trace {Natural ht of col: \the \ht\@outputbox}%
545 \f@trace {\string \@colht: \the \@colht}%
546 \f@trace {Pageshrink added: \the \pageshrink}%
547 \f@trace {Hence, space added: \the \@tempdima}%
548 {/trace}
549 \setbox\@outputbox \vbox to \@colht {%
550 %
551 \boxmaxdepth \maxdepth
552 \unvbox\@outputbox
553 \vskip \@tempdima
554 \textbottom
555 }%

```

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.

```

555 \else
556 \advance \@tempdima -\ht\@kludgeins
557 {*trace}
558 \f@trace {Natural ht of col: \the \ht\@outputbox}%
559 \f@trace {\string \@colht: \the \@colht}%
560 \f@trace {Extra size added: -\the \ht \@kludgeins}%
561 \f@trace {Hence, height of inner box: \the \@tempdima}%
562 \f@trace {Max? pageshrink available: \the \pageshrink}%
563 {/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.

```

564 \setbox \@outputbox \vbox to \@colht {%
565 \vbox to \@tempdima {%

```

```

566 \unvbox\@outputbox
567 \@textbottom}%
568 \vss}%
569 \fi

```

Finally we need to explicitly make the insert box void.

```

570 {\setbox \@tempboxa \box \@kludgeins}%
571 {*trace}
572 \f1@trace {kludgeins box made void}%
573 {/trace}
574 }
575 {/2ekernel | ftrace}

```

(*End of definition for \@makespecialcolbox.*)

\@texttop These do nothing as a default.  
\@textbottom  
576 {\*2ekernel}
577 \let \@texttop \relax
578 \let \@textbottom \relax

(*End of definition for \@texttop and \@textbottom.*)

\@resetactivechars RmS 93/09/06: added hook to protect against certain active characters in the output routine. Default checks are for active space and end-of-line.

```

579 \def\@activechar@info #1{%
580 \clatex@info@no@line {Active #1 character found while
581 output routine is active
582 \MessageBreak
583 This may be a bug in a package file
584 you are using}%
585 }

```

Do not put any spaces in this next bit!

```

586 \begingroup
587 \obeylines\obeyspaces%
588 \catcode`\'\active%
589 \gdef\@resetactivechars{%
590 \def^~M{\@activechar@info{EOL}\space}%
591 \def {\@activechar@info{space}\space}%
592 \let'\active@math@prime}%
593 \endgroup

```

(*End of definition for \@resetactivechars and \@activechar@info.*)

\@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 \writes are handled correctly.

Within shipout's vbox it is reset to its default value, \relax.

Resetting it to its default value after the shipout has been completed (and the contents of the writes have been expanded) must be done by use of \aftergroup. This

is because it must have the value `\relax` before macros coming from other uses of `\aftergroup` within this box are expanded.

Putting this into the `\aftergroup` token list does not affect the definition used in expanding the `\writes` because the aftergroup token list is only constructed when popping the save-stack, it is not expanded until after the shipout is completed.

Question: should things from an `\aftergroup` within the shipped out box be executed in the environment set up for the writes, or after it finishes?

A lot of this code has been in-lined to prevent mis-use of internal commands as hooks.

```
594 {/2ekernel}
595 {latexrelease}\IncludeInRelease{2017/04/15}%
596 {latexrelease} {\@outputpage}{Reset language for hyphenation}%
597 {*2ekernel | latexrelease}
598 \def\@outputpage{%
```

The `\endgroup` is put in by `\aftergroup`.

```
599 \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`.

```
600 \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.

```
601 \language\document@default@language
```

The `\catcode`\\ = 10` was removed as it was considered useless (presumably because nothing gets tokenized during shipout).

This was put in as some error produced active spaces in a mark, I think.

Why was the hyphen reset?

```
602 \@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.

```
603 \global\let\@if@newlist\if@newlist
604 \global@\newlistfalse
```

This next hook replaces the following:

```
\let\-\@dischyp
\let'\@acci\let`\@acci\let\=\@accii
\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\normalineskip

\baselineskip\normalbaselineskip

\sloppy

```

605 \parboxrestore

... to here was in the command \writeshipout.

```

606 \shipout \vbox{%
607 \set@typeset@protect
608 \aftergroup \endgroup

```

Correct? or just restore by ending the group?

609 \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 \cshipoutsetup.

```

610 \if@specialpage
611 \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
612 \fi
613 \if@twoside
614 \ifodd\count\z@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
615 \let\@themargin\oddsidemargin
616 \else \let\@thehead\@evenhead
617 \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
618 \fi
619 \fi

```

The rest was always inside the box.

RmS 91/08/15: added this line:

620 \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.

621 \normalsize

Reset the space factors.

622 \normalsfcodes

Reset these here (previously reset separately for head and foot)

```

623 \let\label\@gobble
624 \let\index\@gobble
625 \let\glossary\@gobble

```

626 \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@

... to here was in the command \shipoutsetup.

```
627 \begingroup
628 \vskip \topmargin
629 \moveleft\@themargin \vbox {%
630 \setbox\@tempboxa \vbox to\headheight{%
631 \vfil
632 \color@hbox
633 \normalcolor
634 \hb@xt@\textwidth{\@thehead}%
635 \color@endbox
636 }%
637 \dp\@tempboxa \z@
638 \box\@tempboxa
639 \vskip \headsep
640 \box\@outputbox
641 \baselineskip \footskip
642 \color@hbox
643 \normalcolor
644 \hb@xt@\textwidth{\@thefoot}%
645 \color@endbox
646 }%
647 }%
```

\endgroup now inserted by \aftergroup

```
Restore \if@newlist
648 \global\let\if@newlist\@if@newlist
649 \global \colht \textheight
650 \stepcounter{page}%
651 }
```

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.

```
651 \let\firstmark\botmark
652 }
653 </2ekernel | latexrelease>
654 <latexrelease>\EndIncludeInRelease
655 <latexrelease>\IncludeInRelease{0000/00/00}%
656 <latexrelease> {\@outputpage}{Reset language for hyphenation}%
657 <latexrelease>\def\@outputpage{%
658 <latexrelease>\begingroup
659 <latexrelease> \let \protect \noexpand
660 <latexrelease> \resetactivechars
661 <latexrelease> \global\let\@if@newlist\if@newlist
662 <latexrelease> \global\@newlistfalse
663 <latexrelease> \parboxrestore
664 <latexrelease> \shipout \vbox{%
665 <latexrelease> \set@typeset@protect
666 <latexrelease> \aftergroup \endgroup
667 <latexrelease> \aftergroup \set@typeset@protect
668 <latexrelease> \if@specialpage
669 <latexrelease> \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
670 <latexrelease> \fi
```

```

671 <{latexrelease}> \if@twoside
672 <{latexrelease}> \ifodd\countz@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
673 <{latexrelease}> \let\@themargin\oddsidemargin
674 <{latexrelease}> \else \let\@thehead\@evenhead
675 <{latexrelease}> \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
676 <{latexrelease}> \fi
677 <{latexrelease}> \fi
678 <{latexrelease}> \reset@font
679 <{latexrelease}> \normalsize
680 <{latexrelease}> \normalsfcodes
681 <{latexrelease}> \let\label\gobble
682 <{latexrelease}> \let\index\gobble
683 <{latexrelease}> \let\glossary\gobble
684 <{latexrelease}> \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
685 <{latexrelease}> \vskip \topmargin \moveright\@themargin \vbox {%
686 <{latexrelease}> \begindvi
687 <{latexrelease}> \vskip \topmargin
688 <{latexrelease}> \moveright\@themargin \vbox {%
689 <{latexrelease}> \setbox\@tempboxa \vbox to\headheight{%
690 <{latexrelease}> \vfil
691 <{latexrelease}> \color\hbox
692 <{latexrelease}> \normalcolor
693 <{latexrelease}> \hb@xt@\textwidth{\@thehead}%
694 <{latexrelease}> \color\endbox
695 <{latexrelease}> }%
696 <{latexrelease}> \dp\@tempboxa \z@
697 <{latexrelease}> \box\@tempboxa
698 <{latexrelease}> \vskip \headsep
699 <{latexrelease}> \box\@outputbox
700 <{latexrelease}> \baselineskip \footskip
701 <{latexrelease}> \color\hbox
702 <{latexrelease}> \normalcolor
703 <{latexrelease}> \hb@xt@\textwidth{\@thefoot}%
704 <{latexrelease}> \color\endbox
705 <{latexrelease}> }%
706 <{latexrelease}> }%
707 <{latexrelease}> \global\let\if@newlist\@@if@newlist
708 <{latexrelease}> \global \colht \textheight
709 <{latexrelease}> \stepcounter{page}%
710 <{latexrelease}> \let\firstmark\botmark
711 <{latexrelease}> }
712 <{latexrelease}> \EndIncludeInRelease
713 <{*2ekernel}>

```

(End of definition for \outputpage, \shipoutsetup, and \writesetup.)

\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.

```

714 \def \begindvi{%
715 \unvbox \begindvibox
716 \global\let \begindvi \empty
717 }

```

(End of definition for \begindvi.)

\@combinefloats The \boxmaxdepth setting here was not made local to a box so was dangerous. It is needed only within the box made by \@cflt (and not normally even there), so it has been moved there; this also agrees with the original pseudocode.

```

718 \def \@combinefloats {%
719 % \boxmaxdepth \maxdepth
720 \ifx \@toplist\@empty \else \@cflt \fi
721 \ifx \@botlist\@empty \else \@cflb \fi
722 }

723 \def \@cflt{%
724 \let \@elt \@comflelt
725 \setbox\@tempboxa \vbox{}%
726 \@toplist
727 \setbox\@outputbox \vbox{%
728 \boxmaxdepth \maxdepth
729 \unvbox\@tempboxa
730 \vskip -\floatsep
731 \topfigrule
732 \vskip \textfloatsep
733 \unvbox\@outputbox
734 }%
735 \let\@elt\relax
736 \xdef\@freelist{\@freelist\@toplist}%
737 \global\let\@toplist\@empty
738 }

739 \def \@cflb {%
740 \let\@elt\@comflelt
741 \setbox\@tempboxa \vbox{}%
742 \@botlist
743 \setbox\@outputbox \vbox{%
744 \unvbox\@outputbox
745 \vskip \textfloatsep
746 \botfigrule
747 \unvbox\@tempboxa
748 \vskip -\floatsep
749 }%
750 \let\@elt\relax
751 \xdef\@freelist{\@freelist\@botlist}%
752 \global \let \@botlist\@empty
753 }

```

(End of definition for \@combinefloats, \@cflt, and \@cflb.)

```

\@comflelt
\@comdblflflelt
754 \def\@comflelt#1{\setbox\@tempboxa
755 \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
756 \def\@comdblflflelt#1{\setbox\@tempboxa
757 \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
758 \def \@combinedblfloats{%
759 \ifx \@dbltoplist \@empty
760 \else
761 \setbox\@tempboxa \vbox{}%
762 \let \@elt \@comdblflflelt

```

```

763 \@dbltoplist
764 \let \@elt \relax
765 \xdef \@freelist {\@freelist\@dbltoplist}%
766 \global \let \@dbltoplist \empty
767 \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.

```

768 {%\boxmaxdepth\maxdepth %% probably not needed, CAR
769 \unvbox\@tempboxa\vskip-\dblfloatsep

```

Here we need different typesetting if the top float comes from `\@topnewpage`.

```

770 \ifnum \@dbltopnum>\m@ne
771 \dblfigrule
772 \fi
773 \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.

```

774 \unvbox\@outputbox
775 }%
776 \fi
777 }
778 </2ekernel>

```

(End of definition for `\@comflelt`, `\@comdblfllelt`, and `\@combinedblfloats`.)

`\@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`.

```

779 <*2ekernel | fltrace>
780 \def \@startcolumn {%
781 \global \@colroom \@colht
782 \@tryfcolumn \@deferlist
783 \if@fcolmade
784 (*trace)
785 \fl@trace{PAGE: float \if@twocolumn column \else page \fi
786 completed}%
787 </trace>
788 \else
789 \begingroup
790 \let \reserved@b \@deferlist
791 \global \let \@deferlist \empty
792 \let \@elt \@scolelt
793 \reserved@b
794 \endgroup
795 \fi
796 }

```

This one does not need to set \@colht.

```
797 〈/2ekernel | fltrace〉
798 〈latexrelease | fltrace〉\IncludeInRelease{2015/01/01}%">
799 〈latexrelease | fltrace〉 {\@startdblcolumn}{float order in 2-column}%">
800 〈*2ekernel | latexrelease | fltrace〉
801 \def \@startdblcolumn {%802 \@tryfcolumn \@deferlist
803 \if@fcolmade
804 \fl@trace{PAGE: double float page completed}%">
805 \else
806 \begingroup
807 \let \reserved@b \@deferlist
808 \global \let \@deferlist \empty
809 \let \@elt \@sdblcolelt
810 \reserved@b
811 \endgroup
812 \fi
813 }%
814 〈/2ekernel | latexrelease | fltrace〉
815 〈latexrelease | fltrace〉\EndIncludeInRelease
816 〈latexrelease | fltrace〉\IncludeInRelease{0000/00/00}%">
817 〈latexrelease | fltrace〉 {\@startdblcolumn}{float order in 2-column}%">
818 〈latexrelease | fltrace〉\def \@startdblcolumn {%
```

Not needed since this always comes after \@outputpage:

```
819 〈latexrelease | fltrace〉% \global \@colht \textheight
820 〈latexrelease | fltrace〉 \@tryfcolumn \@dbldeferlist
821 〈latexrelease | fltrace〉 \if@fcolmade
822 〈*trace〉
823 〈latexrelease | fltrace〉 \fl@trace{PAGE: double float page completed}%">
824 〈/trace〉
825 〈latexrelease | fltrace〉 \else

826 〈latexrelease | fltrace〉 \begingroup
827 〈latexrelease | fltrace〉 \let \reserved@b \@dbldeferlist
828 〈latexrelease | fltrace〉 \global \let \@dbldeferlist \empty
829 〈latexrelease | fltrace〉 \let \@elt \@sdblcolelt
830 〈latexrelease | fltrace〉 \reserved@b
831 〈latexrelease | fltrace〉 \endgroup
832 〈latexrelease | fltrace〉 \fi
833 〈latexrelease | fltrace〉}%">
834 〈latexrelease | fltrace〉\EndIncludeInRelease
835 〈*2ekernel | fltrace〉
```

(End of definition for \@startcolumn and \@startdblcolumn.)

\@tryfcolumn Now tests if its list is empty before any further exertion.

```
836 \def \@tryfcolumn #1{
837 \global \@fcolmadefalse
838 \ifx #1\empty
839 \else
840 \fl@trace{PAGE: try float \if@twocolumn column/page\else page\fi
841 ---\string #1}%">
842 }
```

```

843 \f1@trace{---- \string #1}%
844
```

`</trace>`

```

845 \xdef\@trylist{\#1}%
846 \global\let\@failedlist\@empty
847 \begingroup
848 \let\@elt\@xtryfc\@trylist
849 \endgroup
850 \if@fcolmade
851 \@vtryfc\#1%
852 \fi
853 \fi
854 }
855
```

`</2ekernel | ftrace>`

(End of definition for `\@tryfc`.)

```

856
```

 `$\{*2ekernel\}$` 

`\@scolelt`

```

857 \def\@scolelt#1{\def\@currbox{\#1}\@addtonextcol}

```

(End of definition for `\@scolelt`.)

`\@sdblcolelt`

```

858 \def\@sdblcolelt#1{\def\@currbox{\#1}\@addtobdblcol}

```

(End of definition for `\@sdblcolelt`.)

`\@vtryfc`

```

859 \def\@vtryfc\#1{%
860 \global\setbox\@outputbox\vbox{}%
861 \let\@elt\@wtryfc
862 \if@flsucceed
863 \global\setbox\@outputbox\vbox{to\@colht}{%
864 \vskip\@fptop
865 \vskip-\@fpsep
866 \unvbox\@outputbox
867 \vskip\@fpbot}%
868 \let\@elt\relax
869 \xdef\#1{\@failedlist\@flfail}%
870 \xdef\@freelist{\@freelist\@flsucceed}}

```

(End of definition for `\@vtryfc`.)

`\@wtryfc`

```

871 \def\@wtryfc\#1{%
872 \global\setbox\@outputbox\vbox{%
873 \unvbox\@outputbox
874 \vskip\@fpsep
875 \box\#1}}

```

(End of definition for `\@wtryfc`.)

```

\@xtryfc
 876 </2ekernel>
 877 <|latexrelease|>\IncludeInRelease{2015/01/01}{\@xtryfc}%
 878 <|latexrelease|> {float order in 2-column}%
 879 <*2ekernel | latexrelease>
 880 \def\@xtryfc #1{%
 881 \@next\reserved@a\@trylist{}{}%
 882 \@currtype \count #1%
 883 \divide\@currtype\@xxxii
 884 \multiply\@currtype\@xxxii
 885 \or\@bitor \@currtype \@failedlist
 886 \@testfp #1%
 887 \@testwrongwidth #1%
 888 \ifdim \ht #1>\@colht
 889 \@testtrue
 890 \fi
 891 \if@test
 892 \cons\@failedlist #1%
 893 \else
 894 \@ytryfc #1%
 895 \fi}%
 896 </2ekernel | latexrelease>
 897 <|latexrelease|>\EndIncludeInRelease
 898 <|latexrelease|>\IncludeInRelease{0000/00/00}{\@xtryfc}%
 899 <|latexrelease|> {float order in 2-column}%
 900 <|latexrelease|>\def\@xtryfc #1{%
 901 <|latexrelease|> \@next\reserved@a\@trylist{}{}%
 902 <|latexrelease|> \@currtype \count #1%
 903 <|latexrelease|> \divide\@currtype\@xxxii
 904 <|latexrelease|> \multiply\@currtype\@xxxii
 905 <|latexrelease|> \or\@bitor \@currtype \@failedlist
 906 <|latexrelease|> \@testfp #1%
 907 <|latexrelease|> \ifdim \ht #1>\@colht
 908 <|latexrelease|> \@testtrue
 909 <|latexrelease|> \fi
 910 <|latexrelease|> \if@test
 911 <|latexrelease|> \cons\@failedlist #1%
 912 <|latexrelease|> \else
 913 <|latexrelease|> \@ytryfc #1%
 914 <|latexrelease|> \fi}%
 915 <|latexrelease|>\EndIncludeInRelease
 916 <*2ekernel>

```

(End of definition for \@xtryfc.)

```

\@ytryfc
 917 \def\@ytryfc #1{%
 918 \begingroup
 919 \gdef\@flsucceed{\@elt #1}%
 920 \global\let\@flfail\empty
 921 \tempdima\ht #1%
 922 \let\@elt\@ztryfc
 923 \@trylist

```

```

924 \ifdim \@tempdima >\@fpmin
925 \global\@fcolmadetrue
926 \else
927 \@cons\@failedlist #1%
928 \fi
929 \endgroup
930 \if@fcolmade
931 \let\@elt\@gobble
932 \fi}

(End of definition for \ztryfc.)
```

```

\@ztryfc
933 〈/2ekernel〉
934 〈latexrelease〉\IncludeInRelease{2015/01/01}{\ztryfc}%
935 〈latexrelease〉
936 〈*2ekernel | latexrelease〉
937 \def\ztryfc #1{%
938 \@tempcnta\count #1%
939 \divide\@tempcnta\@xxxii
940 \multiply\@tempcnta\@xxxii
941 \@bitor \@tempcnta {\@failedlist \@flfail}%
942 \@testfp #1%
943 not in fixfloats?
944 \@testwrongwidth #1%
945 \@tempdimb\@tempdima
946 \advance\@tempdimb\ht #1%
947 \advance\@tempdimb\@fpsep
948 \ifdim \@tempdimb >\@colht
949 \@testtrue
950 \fi
951 \if@test
952 \@cons\@flfail #1%
953 \else
954 \@cons\@flsucceed #1%
955 \fi}%
956 〈/2ekernel | latexrelease〉
957 〈latexrelease〉\EndIncludeInRelease
958 〈latexrelease〉\IncludeInRelease{0000/00/00}{\ztryfc}%
959 〈latexrelease〉
960 〈latexrelease〉\def\ztryfc #1{%
961 \@tempcnta \count#1%
962 \divide\@tempcnta\@xxxii
963 \multiply\@tempcnta\@xxxii
964 \@bitor \@tempcnta {\@failedlist \@flfail}%
965 \@testfp #1%
966 \@tempdimb\@tempdima
967 \advance\@tempdimb\ht#1%
968 \advance\@tempdimb\@fpsep
969 \ifdim \@tempdimb >\@colht
970 \@testtrue
971 \fi

```

```

972 <{latexrelease} \if@test
973 <{latexrelease} \@cons\@flfail #1%
974 <{latexrelease} \else
975 <{latexrelease} \@cons\@flsucceed #1%
976 <{latexrelease} \@tempdima\@tempdimb
977 <{latexrelease} \fi}%
978 <{latexrelease}\EndIncludeInRelease

```

(End of definition for \@ztryfc.)

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.

```

979 <{*2ekernel | fltrace}
980 \def \@addtobot {%
981 <{*trace}
982 \fl@trace{***Start addtobot}%
983 </trace}
984 \@getfpsbit 4\relax
985 <{*trace}
986 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi bot:
987 \the \@fpstype}%
988 </trace}
989 \ifodd \@tempcnta
990 \@flsetnum \@botnum
991 \ifnum \@botnum>\z@
992 \@tempswafalse
993 \@flcheckspace \@botroom \@botlist
994 \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?

```

995 \global \maxdepth \z@
996 \@flupdates \@botnum \@botroom \@botlist
997 <{*trace}
998 \fl@trace{colroom (after-bot) = \the \@colroom}%
999 \fl@trace{colnum (after-bot) = \the \@colnum}%
1000 \fl@trace{botnum (after-bot) = \the \@botnum}%
1001 \fl@trace{***Success: bot}%
1002 </trace}
1003 \Qinserttrue
1004 \fi
1005 <{*trace}
1006 \else
1007 \fl@trace{Fail: botnum = \the \@botnum:
1008 fpstype \the \@fpstype=ORD?}%
1009 \ifnum \@fpstype<\sixt@n
1010 \fl@trace{ERROR: !b float not successful (addtobot)}%
1011 \fi
1012 </trace}
1013 \fi
1014 \fi
1015 }

```

(End of definition for \@addtobot.)

\@addtotoporbot Lots of changes.

```
1016 \def \@addtotoporbot {%
1017 <*trace>
1018 \f1@trace{***Start addtotoporbot}%
1019 </trace>
1020 \@getfpsbit \tw@
1021 <*trace>
1022 \f1@trace{fpstype \ifodd \@tempcnta OK \else not \fi top:
1023 \the \@fpstype}%
1024 </trace>
1025 \ifodd \@tempcnta
1026 \@flsetnum \@topnum
1027 \ifnum \@topnum>\z@
1028 \@tempswafalse
1029 \@flcheckspace \@toproom \@toplist
1030 \if@tempswa
1031 \obitor\@currtype{\@midlist\@botlist}%
1032 <*trace>
1033 \f1@trace{(mid+bot)list: \@midlist, \@botlist:
1034 (addtotoporbot-before)}%
1035 </trace>
1036 \if@test
1037 <*trace>
1038 \f1@trace{type already on list: mid or bot---sent to addtobot}%
1039 </trace>
1040 \else
1041 \@flupdates \@topnum \@toproom \@toplist
1042 <*trace>
1043 \f1@trace{colroom (after-top) = \the \@colroom}%
1044 \f1@trace{colnum (after-top) = \the \@colnum}%
1045 \f1@trace{topnum (after-top) = \the \@topnum}%
1046 \f1@trace{***Success: top}%
1047 </trace>
1048 \@inserttrue
1049 \fi
1050 \fi
1051 <*trace>
1052 \else
1053 \f1@trace{Fail: topnum = \the \@topnum: fpstype
1054 \the \@fpstype=ORD?}%
1055 \ifnum \@fpstype<\sixt@n
1056 \f1@trace{ERROR: !t float not successful (addtotoporbot)}%
1057 \fi
1058 </trace>
1059 \fi
1060 \fi
1061 \if@insert
1062 \else
1063 <*trace>
1064 \f1@trace{sent to addtobot (addtotoporbot)}%
1065 </trace>
1066 \@addtobot
1067 \fi
1068 }
```

```

1069 ⟨/2ekernel | fltrace⟩

(End of definition for \@addtotoporbot.)
```

\@addtocurcol Lots of changes.

```

1070 ⟨| latexrelease | fltrace | flafter⟩\IncludeInRelease{2015/01/01}%
1071 ⟨| latexrelease | fltrace | flafter⟩ {⟨\@addtocurcol⟩{float order in 2-column}%
1072 ⟨*2ekernel | latexrelease | fltrace | flafter⟩
1073 \def \@addtocurcol {%
1074 ⟨*trace⟩
1075 \fl@trace{***Start addtocurcol}%
1076 ⟨/trace⟩
1077 \cinsertfalse
1078 \setfloattypecounts
1079 \ifnum \fpstype=8
1080 ⟨*trace⟩
1081 \fl@trace{fpstype !p only (addtocurcol): \the \fpstype = 8?}%
1082 ⟨/trace⟩
1083 \else
1084 \ifnum \fpstype=24
1085 ⟨*trace⟩
1086 \fl@trace{fpstype p only (addtocurcol): \the \fpstype = 24?}%
1087 ⟨/trace⟩
1088 \else
1089 \cflsettextmin
```

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.

```

1090 ⟨*trace⟩
1091 \fl@trace{textfloatsheight (before) = \the \textfloatsheight}%
1092 ⟨/trace⟩
1093 \advance \textmin \textfloatsheight
1094 \reqcolroom \pageht
```

This line must be removed since \specialoutput changed.

```

1095 % \advance \reqcolroom \pagedp
1096 ⟨*trace⟩
1097 \fl@trace{textmin + textfloatsheight: \the \textmin}%
1098 \fl@trace{page-so-far: \the \reqcolroom}%
1099 ⟨/trace⟩
1100 \ifdim \textmin>\reqcolroom
1101 \reqcolroom \textmin
1102 ⟨*trace⟩
1103 \fl@trace{ORD? textmin being used}%
1104 ⟨/trace⟩
1105 \fi
1106 \advance \reqcolroom \ht\currbox
1107 ⟨*trace⟩
1108 \fl@trace{float size = \the \ht \currbox (addtocurcol)}%
1109 \fl@trace{colroom = \the \colroom (addtocurcol)}%
1110 \fl@trace{reqcolroom = \the \reqcolroom (addtocurcol)}%
1111 ⟨/trace⟩
```

```

1112 \ifdim \@colroom>\@reqcolroom
1113 \@flsetnum \@colnum
1114 \ifnum \@colnum>\z@
1115 \@bitor\@currtype\@deferlist

```

We need to defer the float also if its width doesn't fit.

```

1116 \@testwrongwidth\@currbox
1117 <*trace>
1118 \fl@trace{deferlist: \@deferlist: (addtocurcol-before)}%
1119 </trace>
1120 \if@test
1121 <*trace>
1122 \fl@trace{type already on list: defer (addtocurcol)}%
1123 </trace>
1124 \else
1125 \@bitor\@currtype\@botlist
1126 <*trace>
1127 \fl@trace{botlist: \@botlist: (addtocurcol-before)}%
1128 </trace>
1129 \if@test
1130 <*trace>
1131 \fl@trace{type already on list: bot---sent to addtobot}%
1132 </trace>
1133 \@addtobot
1134 \else
1135 <*trace>
1136 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi
1137 here: \the \@fpstype}%
1138 </trace>
1139 \ifodd \count\@currbox
1140 \advance \@reqcolroom \intextsep
1141 \ifdim \@colroom>\@reqcolroom
1142 \global \advance \@colnum \m@ne
1143 \global \advance \@textfloatsheight \ht\@currbox

```

This may sometimes give an overestimate.

```

1144 \global \advance \@textfloatsheight 2\intextsep
1145 \@cons \@midlist \@currbox
1146 <*trace>
1147 \fl@trace{***Success: here}%
1148 \fl@trace{textfloatsheight (after-here) =
1149 \the \@textfloatsheight}%
1150 \fl@trace{colnum (after-here) = \the \@colnum}%
1151 </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.

```

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).

```

1152 \if@nobreak
1153 \nobreak
1154 \nobreakfalse
1155 \everypar{}%
1156 \else
1157 \addpenalty \interlinepenalty
1158 \fi
1159 \vskip \intextsep
1160 \box@\currbox
1161 \penalty\interlinepenalty
1162 \vskip\intextsep
1163 \ifnum\outputpenalty <- \Mii \vskip -\parskip\fi

```

Typesetting ends here.

```

1164 \outputpenalty \z@
1165 \inserttrue
1166 {*trace}
1167 \else
1168 \f@trace{Fail---no room at 2nd test of colroom
1169 (addtocorcol \string\intextsep)}%
1170
```

 $\{/trace\}$ 

```

1171 \fi
1172 \fi
1173 \if@insert
1174 \else

```

Next set of docstrip guards are a bit weird, essentially \c@addtotopbot ends up inside the kernel and the `ftrace` package and \c@addtobot shows up in the `flafter` package. Guess that could have been done a bit more obvious :-)

```

1175 {*2ekernel | ftrace | latexrelease}
1176 {*trace}
1177 \f@trace{not here: sent to addtotopbot}%
1178
```

 $\{/trace\}$ 

```

1179 \c@addtotopbot
1180
```

 $\{/2ekernel | ftrace | latexrelease\}$ 

```

1181 {*!2ekernel&!ftrace&!latexrelease}
1182 {*trace}
1183 \f@trace{not here: sent to addtobot}%
1184
```

 $\{/trace\}$ 

```

1185 \c@addtobot
1186
```

 $\{/!2ekernel&!ftrace&!latexrelease\}$ 

```

1187 \fi
1188 \fi
1189 \fi
1190
```

 $\{*trace\}$ 

```

1191 \else
1192 \f@trace{Fail: colnum = \the \c@colnum:
1193 fpstype \the \c@fpstype=ORD?}%
1194 \ifnum \c@fpstype<\sixt@n
1195 \f@trace{ERROR: BANG float not successful (addtocurcol)}%
1196 \fi

```

```

1197 〈/trace〉
1198 \fi
1199 〈*trace〉
1200 \else
1201 \fl@trace{Fail---no room: fl box ht: \the \ht \currbox
1202 (addtocurcol)}%
1203 〈/trace〉
1204 \fi
1205 \fi
1206 \fi
1207 \if@insert
1208 \else
1209 \@resethfps
1210 〈*trace〉
1211 \fl@trace{put on deferlist (addtocurcol)}%
1212 〈/trace〉
1213 \@cons\@deferlist\@currbox
1214 〈*trace〉
1215 \fl@trace{deferlist: \@deferlist: (addtocurcol-after)}%
1216 〈/trace〉
1217 \fi
1218 }%
1219 (./2ekernel | latexrelease | fltrace | flafter)
1220 (latexrelease | fltrace | flafter)\EndIncludeInRelease
1221 (latexrelease | fltrace | flafter)\IncludeInRelease{0000/00/00}%
1222 (latexrelease | fltrace | flafter) {\@addtocurcol}{float order in 2-column}%
1223 (latexrelease | fltrace | flafter)\def \@addtocurcol {%
1224 〈*trace〉
1225 (latexrelease | fltrace | flafter) \fl@trace{***Start addtocurcol}%
1226 〈/trace〉
1227 (latexrelease | fltrace | flafter) \@insertfalse
1228 (latexrelease | fltrace | flafter) \@setfloattypecounts
1229 (latexrelease | fltrace | flafter) \ifnum \@fpstype=8
1230 〈*trace〉
1231 (latexrelease | fltrace | flafter) \fl@trace{fpstype !p only (addtocurcol):
1232 (latexrelease | fltrace | flafter) \the \@fpstype = 8?}%
1233 〈/trace〉
1234 (latexrelease | fltrace | flafter) \else
1235 (latexrelease | fltrace | flafter) \ifnum \@fpstype=24
1236 〈*trace〉
1237 (latexrelease | fltrace | flafter) \fl@trace{fpstype p only (addtocurcol):
1238 (latexrelease | fltrace | flafter) \the \@fpstype = 24?}%
1239 〈/trace〉
1240 (latexrelease | fltrace | flafter) \else
1241 (latexrelease | fltrace | flafter) \@flesstextmin

```

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.

```

1242 〈*trace〉
1243 (latexrelease | fltrace | flafter) \fl@trace{textfloatsheight (before) =
1244 (latexrelease | fltrace | flafter) \the \@textfloatsheight}%
1245 〈/trace〉

```

```

1246 〈latexrelease | fltrace | flafter〉 \advance \@textmin \@textfloatsheight
1247 〈latexrelease | fltrace | flafter〉 \@reqcolroom \@pageht
This line must be removed since \@specialoutput changed.
1248 % \advance \@reqcolroom \@pagedp
1249 〈*trace〉
1250 〈latexrelease | fltrace | flafter〉 \fl@trace{textmin + textfloatsheight:
1251 〈latexrelease | fltrace | flafter〉 \the \@textmin}%
1252 〈latexrelease | fltrace | flafter〉 \fl@trace{page-so-far: \the \@reqcolroom}%
1253 〈latexrelease | fltrace | flafter〉
1254 〈/trace〉
1255 〈latexrelease | fltrace | flafter〉
1256 〈latexrelease | fltrace | flafter〉
1257 〈*trace〉
1258 〈latexrelease | fltrace | flafter〉
1259 〈/trace〉
1260 〈latexrelease | fltrace | flafter〉
1261 〈latexrelease | fltrace | flafter〉
1262 〈*trace〉
1263 〈latexrelease | fltrace | flafter〉
1264 〈latexrelease | fltrace | flafter〉
1265 〈latexrelease | fltrace | flafter〉
1266 〈latexrelease | fltrace | flafter〉
1267 〈latexrelease | fltrace | flafter〉
1268 〈latexrelease | fltrace | flafter〉
1269 〈/trace〉
1270 〈latexrelease | fltrace | flafter〉
1271 〈latexrelease | fltrace | flafter〉
1272 〈latexrelease | fltrace | flafter〉
1273 〈latexrelease | fltrace | flafter〉
1274 〈*trace〉
1275 〈latexrelease | fltrace | flafter〉
1276 〈latexrelease | fltrace | flafter〉
1277 〈/trace〉
1278 〈latexrelease | fltrace | flafter〉
1279 〈*trace〉
1280 〈latexrelease | fltrace | flafter〉
1281 〈latexrelease | fltrace | flafter〉
1282 〈/trace〉
1283 〈latexrelease | fltrace | flafter〉
1284 〈latexrelease | fltrace | flafter〉
1285 〈*trace〉
1286 〈latexrelease | fltrace | flafter〉
1287 〈latexrelease | fltrace | flafter〉
1288 〈/trace〉
1289 〈latexrelease | fltrace | flafter〉
1290 〈*trace〉
1291 〈latexrelease | fltrace | flafter〉
1292 〈latexrelease | fltrace | flafter〉
1293 〈/trace〉
1294 〈latexrelease | fltrace | flafter〉
1295 〈latexrelease | fltrace | flafter〉
1296 〈*trace〉
1297 〈latexrelease | fltrace | flafter〉
1298 〈latexrelease | fltrace | flafter〉 \advance \@textmin \@textfloatsheight
 \@reqcolroom \@pageht
\ifdim \@textmin>\@reqcolroom
 \@reqcolroom \@textmin
 \fl@trace{ORD? textmin being used}%
\fi
\advance \@reqcolroom \ht\@currbox
\fl@trace{float size =
 \the \ht \@currbox (addtocurcol)}%
\fl@trace{colroom =
 \the \@colroom (addtocurcol)}%
\fl@trace{reqcolroom =
 \the \@reqcolroom (addtocurcol)}%
\ifdim \@colroom>\@reqcolroom
 \@flsetnum \@colnum
 \ifnum \@colnum>z@
 \@bitor\@currtype\@deferlist
 \fl@trace{deferlist:
 \@deferlist: (addtocurcol-before)}%
 \if@test
 \fl@trace{type already on list:
 defer (addtocurcol)}%
 \else
 \@bitor\@currtype\@botlist
 \fl@trace{botlist: \@botlist:
 (addtocurcol-before)}%
 \if@test
 \fl@trace{type already on list:
 bot---sent to addtobot}%
 \@addtobot
 \else
 \fl@trace{fpstype
 \ifodd \@tempcnta OK \else not \fi

```

```

1299 <{latexrelease | fltrace | flafter}>
1300 </trace>
1301 <{latexrelease | fltrace | flafter}>
1302 <{latexrelease | fltrace | flafter}>
1303 <{latexrelease | fltrace | flafter}>
1304 <{latexrelease | fltrace | flafter}>
1305 <{latexrelease | fltrace | flafter}>
1306 <{latexrelease | fltrace | flafter}>
1307 <{latexrelease | fltrace | flafter}>
1308 <{latexrelease | fltrace | flafter}>
1309 <{latexrelease | fltrace | flafter}>
1310 <{*trace}>
1311 <{latexrelease | fltrace | flafter}>
1312 <{latexrelease | fltrace | flafter}>
1313 <{latexrelease | fltrace | flafter}>
1314 <{latexrelease | fltrace | flafter}>
1315 <{latexrelease | fltrace | flafter}>
1316 <{latexrelease | fltrace | flafter}>
1317 </trace>

This may sometimes give an overestimate.

CHANGE TO \addtocurcol:
\penalty\z@ changed to \penalty\interlinepenalty so \samepage works prop-
erly with figure and table environments. (Changed 23 Oct 86)
There is also an \addpenalty\interlinepenalty above.
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).

1318 <{latexrelease | fltrace | flafter}>
1319 <{latexrelease | fltrace | flafter}>
1320 <{latexrelease | fltrace | flafter}>
1321 <{latexrelease | fltrace | flafter}>
1322 <{latexrelease | fltrace | flafter}>
1323 <{latexrelease | fltrace | flafter}>
1324 <{latexrelease | fltrace | flafter}>
1325 <{latexrelease | fltrace | flafter}>
1326 <{latexrelease | fltrace | flafter}>
1327 <{latexrelease | fltrace | flafter}>
1328 <{latexrelease | fltrace | flafter}>
1329 <{latexrelease | fltrace | flafter}>
1330 <{latexrelease | fltrace | flafter}>
1331 <{latexrelease | fltrace | flafter}>

Typesetting ends here.

1332 <{latexrelease | fltrace | flafter}>
1333 <{latexrelease | fltrace | flafter}>
1334 <{*trace}>
1335 <{latexrelease | fltrace | flafter}>
```

```

here: \the \cftpstype}%
\ifodd \count\currbox
\advance \reqcolroom \intextsep
\ifdim \colroom\reqcolroom
\global \advance \colnum \m@ne
\global \advance
\textfloatsheight\ht\currbox
\global \advance
\textfloatsheight 2\intextsep
\cons \midlist \currbox
\ftrace{***Success: here}%
\ftrace{textfloatsheight
(after-here) =
\the \textfloatsheight}%
\ftrace{colnum (after-here) =
\the \colnum}%

```

```

1336 <{latexrelease | fltrace | flafter}> \fl@trace{Fail---no room at 2nd test of colroom
1337 <{latexrelease | fltrace | flafter}> (addtocorcol \string\intextsep)}%
1338 </trace>
1339 <{latexrelease | fltrace | flafter}> \fi
1340 <{latexrelease | fltrace | flafter}> \fi
1341 <{latexrelease | fltrace | flafter}> \if@insert
1342 <{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 :-)

```

1343 <{*2ekernel | fltrace}>
1344 <{*trace}>
1345 <{latexrelease | fltrace | flafter}> \fl@trace{not here: sent to addtotoporbot}%
1346 </trace>
1347 <{latexrelease | fltrace | flafter}> \@addtotoporbot
1348 </2ekernel | fltrace>
1349 <{*!2ekernel&!autoload&!fltrace}>
1350 <{*trace}>
1351 <{latexrelease | fltrace | flafter}> \fl@trace{not here: sent to addtobot}%
1352 </trace>
1353 <{latexrelease | fltrace | flafter}> \@addtobot
1354 </!2ekernel&!autoload&!fltrace>
1355 <{latexrelease | fltrace | flafter}> \fi
1356 <{latexrelease | fltrace | flafter}> \fi
1357 <{latexrelease | fltrace | flafter}> \fi
1358 <{*trace}>
1359 <{latexrelease | fltrace | flafter}> \else
1360 <{latexrelease | fltrace | flafter}> \fl@trace{Fail: colnum = \the \@colnum:
1361 <{latexrelease | fltrace | flafter}> fpstype \the \@fpstype=ORD?}%
1362 <{latexrelease | fltrace | flafter}> \ifnum \@fpstype<\sixt@n
1363 <{latexrelease | fltrace | flafter}> \fl@trace{ERROR: BANG float not successful
1364 <{latexrelease | fltrace | flafter}> (addtocurcol)}%
1365 <{latexrelease | fltrace | flafter}> \fi
1366 </trace>
1367 <{latexrelease | fltrace | flafter}> \fi
1368 <{*trace}>
1369 <{latexrelease | fltrace | flafter}> \else
1370 <{latexrelease | fltrace | flafter}> \fl@trace{Fail---no room: fl box ht:
1371 <{latexrelease | fltrace | flafter}> \the \ht \@currbox (addtocurcol)}%
1372 </trace>
1373 <{latexrelease | fltrace | flafter}> \fi
1374 <{latexrelease | fltrace | flafter}> \fi
1375 <{latexrelease | fltrace | flafter}> \fi
1376 <{latexrelease | fltrace | flafter}> \if@insert
1377 <{latexrelease | fltrace | flafter}> \else
1378 <{latexrelease | fltrace | flafter}> \resethfps
1379 <{*trace}>
1380 <{latexrelease | fltrace | flafter}> \fl@trace{put on deferlist (addtocurcol)}%
1381 </trace>
1382 <{latexrelease | fltrace | flafter}> \@cons\@deferlist\@currbox
1383 <{*trace}>
1384 <{latexrelease | fltrace | flafter}> \fl@trace{deferlist: \@deferlist:
1385 <{latexrelease | fltrace | flafter}> (addtocurcol-after)}%

```

```

1386 </trace>
1387 <|latexrelease | fltrace | flafter> \fi
1388 <|latexrelease | fltrace | flafter> }%
1389 <|latexrelease | fltrace | flafter>\EndIncludeInRelease

```

(End of definition for \@addtocurcol.)

\@addtonextcol Lots of changes.

```

1390 <|latexrelease | fltrace>\IncludeInRelease{2015/01/01}
1391 <|latexrelease | fltrace> {\@addtonextcol}{float order in 2-column}%
1392 <*2ekernel | latexrelease | fltrace>
1393 \def\@addtonextcol{%
1394 \begingroup
1395 <*trace>
1396 \fl@trace{***Start addtonextcol}%
1397 </trace>
1398 \insertfalse
1399 \setfloattypecounts
1400 \ifnum \fpstype=8
1401 <*trace>
1402 \fl@trace{fpstype not curcol: \the \fpstype = 8?}%
1403 </trace>
1404 \else
1405 \ifnum \fpstype=24
1406 <*trace>
1407 \fl@trace{fpstype not curcol: \the \fpstype = 24?}%
1408 </trace>
1409 \else
1410 \flsettextmin
1411 <*trace>
1412 \fl@trace{text-so-far: Opt (top of col)}%
1413 </trace>
1414 \reqcolroom \ht\currbox
1415 <*trace>
1416 \fl@trace{float size: \the \reqcolroom (addtonextcol)}%
1417 </trace>
1418 \advance \reqcolroom \textmin
1419 <*trace>
1420 \fl@trace{colroom = \the \colroom (addtonextcol)}%
1421 \fl@trace{reqcolroom = \the \reqcolroom (addtonextcol)}%
1422 </trace>
1423 \ifdim \colroom>\reqcolroom
1424 \flsetnum \colnum
1425 \ifnum\colnum>\z@
1426 \bitor\currtype\deferlist
1427 <*trace>
1428 \fl@trace{deferlist: \deferlist: (addtonextcol-before)}%
1429 </trace>
1430 \testwidth\currbox
1431 \if@test
1432 <*trace>
1433 \fl@trace{type already on list: defer (addtonextcol)}%
1434 </trace>

```

```

1435 \else
1436 <*trace>
1437 \fl@trace{sent to addtotoporbot (addtonextcol)}%
1438 </trace>
1439 \@addtotoporbot
1440 \fi
1441 <*trace>
1442 \else
1443 \fl@trace{Fail---no room: fl box ht: \the \ht \currbox
1444 (addtonextcol)}%
1445 </trace>
1446 \fi
1447 \fi
1448 \fi
1449 \if@insert
1450 \else
1451 <*trace>
1452 \fl@trace{put back on deferlist (addtonextcol)}%
1453 </trace>
1454 \@cons\@deferlist\@currbox
1455 <*trace>
1456 \fl@trace{deferlist: \@deferlist: (addtonextcol-after)}%
1457 </trace>
1458 \fi
1459 <*trace>
1460 \fl@trace{End of addtonextcol -- locally counts:}%
1461 \fl@trace{col: \the\colnum. top: \the \topnum. bot: \the \botnum.}%
1462 </trace>
1463 \endgroup
1464 <*trace>
1465 \fl@trace{End of addtonextcol -- globally counts:}%
1466 \fl@trace{col: \the\colnum. top: \the \topnum. bot: \the \botnum.}%
1467 </trace>
1468 }%
1469 }%
1470 </2ekernel | latexrelease | fltrace>
1471 <latexrelease | fltrace>\EndIncludeInRelease
1472 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
1473 <latexrelease | fltrace> {\@addtonextcol}{float order in 2-column}%
1474 <latexrelease | fltrace>\def\@addtonextcol{%
1475 <latexrelease | fltrace> \begingroup
1476 <*trace>
1477 <latexrelease | fltrace> \fl@trace{***Start addtonextcol}%
1478 </trace>
1479 <latexrelease | fltrace> \@insertfalse
1480 <latexrelease | fltrace> \@setfloattypecounts
1481 <latexrelease | fltrace> \ifnum \fpstype=8
1482 <*trace>
1483 <latexrelease | fltrace> \fl@trace{fpstype not curcol:
1484 <latexrelease | fltrace> \the \fpstype = 8?}%
1485 </trace>
1486 <latexrelease | fltrace> \else
1487 <latexrelease | fltrace> \ifnum \fpstype=24
1488 <*trace>

```

```

1489 <{latexrelease | fltrace> \fl@trace{fpstype not curcol:
1490 <{latexrelease | fltrace> \the \@fpstype = 24?}%
1491 </trace>
1492 <{latexrelease | fltrace>
1493 <{latexrelease | fltrace>
1494 <{*trace}>
1495 <{latexrelease | fltrace>
1496 </trace>
1497 <{latexrelease | fltrace>
1498 <{*trace}>
1499 <{latexrelease | fltrace>
1500 <{latexrelease | fltrace>
1501 <{latexrelease | fltrace>
1502 </trace>
1503 <{latexrelease | fltrace>
1504 <{*trace}>
1505 <{latexrelease | fltrace>
1506 <{latexrelease | fltrace>
1507 <{latexrelease | fltrace>
1508 <{latexrelease | fltrace>
1509 </trace>
1510 <{latexrelease | fltrace>
1511 <{latexrelease | fltrace>
1512 <{latexrelease | fltrace>
1513 <{latexrelease | fltrace>
1514 <{*trace}>
1515 <{latexrelease | fltrace>
1516 <{latexrelease | fltrace>
1517 </trace>
1518 <{latexrelease | fltrace>
1519 <{*trace}>
1520 <{latexrelease | fltrace>
1521 <{latexrelease | fltrace>
1522 </trace>
1523 <{latexrelease | fltrace>
1524 <{*trace}>
1525 <{latexrelease | fltrace>
1526 <{latexrelease | fltrace>
1527 </trace>
1528 <{latexrelease | fltrace>
1529 <{latexrelease | fltrace>
1530 <{latexrelease | fltrace>
1531 <{*trace}>
1532 <{latexrelease | fltrace>
1533 <{latexrelease | fltrace>
1534 <{latexrelease | fltrace>
1535 </trace>
1536 <{latexrelease | fltrace>
1537 <{latexrelease | fltrace>
1538 <{latexrelease | fltrace>
1539 <{latexrelease | fltrace>
1540 <{latexrelease | fltrace>
1541 <{*trace}>
1542 <{latexrelease | fltrace> \fl@trace{float size:
 \the \@reqcolroom (addtonextcol)}%
 \advance \@reqcolroom \ht\@currbox
 \fl@trace{colroom =
 \the \@colroom (addtonextcol)}%
 \fl@trace{reqcolroom =
 \the \@reqcolroom (addtonextcol)}%
 \ifdim \@colroom>\@reqcolroom
 \@flsetnum \@colnum
 \ifnum\@colnum>\z@
 \@bitor\@currtype\@deferlist
 \fl@trace{deferlist: \@deferlist:
 (addtonextcol-before)}%
 \if@test
 \fl@trace{type already on list:
 defer (addtonextcol)}%
 \else
 \fl@trace{sent to addtotoporbot
 (addtonextcol)}%
 \else
 \fl@trace{Fail---no room: fl box ht:
 \the \ht \@currbox (addtonextcol)}%
 \fi
 \else
 \fl@insert
 \else
 \fl@trace{put back on deferlist}

```

```

1543 <{latexrelease | fltrace} (addtonextcol)%%
1544 </trace>
1545 <{latexrelease | fltrace} \@cons\@deferlist\@currbox
1546 <{*trace}>
1547 <{latexrelease | fltrace} \fl@trace{\deferlist: \@deferlist:
1548 <{latexrelease | fltrace} (addtonextcol-after)}%
1549 </trace>
1550 <{latexrelease | fltrace} \fi
1551 <{*trace}>
1552 <{latexrelease | fltrace} \fl@trace{End of addtonextcol --
1553 <{latexrelease | fltrace} locally counts:}%
1554 <{latexrelease | fltrace} \fl@trace{col: \the \colnum.
1555 <{latexrelease | fltrace} top: \the \topnum. bot: \the \botnum.}%
1556 </trace>
1557 <{latexrelease | fltrace} \endgroup
1558 <{*trace}>
1559 <{latexrelease | fltrace} \fl@trace{End of addtonextcol --
1560 <{latexrelease | fltrace} globally counts:}%
1561 <{latexrelease | fltrace} \fl@trace{col: \the \colnum.
1562 <{latexrelease | fltrace} top: \the \topnum. bot: \the \botnum.}%
1563 </trace>
1564 <{latexrelease | fltrace} }%
1565 <{latexrelease | fltrace}\EndIncludeInRelease

```

(End of definition for \@addtonextcol.)

\@addtobdblcol Lots of changes.

```

1566 <{latexrelease | fltrace}\IncludeInRelease{2015/01/01}%
1567 <{latexrelease | fltrace} {\@addtobdblcol}{float order in 2-column}%
1568 <{*2ekernel | latexrelease | fltrace}
1569 \def\@addtobdblcol{%
1570 \begingroup
1571 <{*trace}>
1572 \fl@trace{***Start addtobdblcol}%
1573 </trace>
1574 \@insertfalse
1575 \@setfloattypecounts
1576 \@getfpsbit \tw@
1577 <{*trace}>
1578 \fl@trace{fpstype \ifodd \tempcnta OK \else not \fi dbltop:
1579 \the \fpstype}%
1580 </trace>
1581 \ifodd\@tempcnta
1582 \@flsetnum \dbltopnum
1583 \ifnum \dbltopnum>\z@
1584 \@tempswafalse
1585 \ifdim \dbltoproom>\ht\@currbox
1586 \@tempswatrue
1587 <{*trace}>
1588 \fl@trace{Space OK: \dbltoproom =
1589 \the \dbltoproom > \the \ht \@currbox
1590 (\dbltoproom)}%
1591 </trace>
1592 \else

```

```

1593 <*trace>
1594 \fl@trace{fpstype: \the \fpstype (addtoblcol)}%
1595 </trace>
1596 \ifnum \fpstype<\sixt@n
1597 <*trace>
1598 \fl@trace{BANG float ignoring \@dbltoproom}%
1599 \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1600 Ht float: \the \ht \currbox-BANG}%
1601 </trace>

```

Need to check that there is room on the page, using the local value of \textmin to make the necessary adjustment to \@dbltoproom.

```

1602 \advance \@dbltoproom \textmin
1603 <*trace>
1604 \fl@trace{Local value of texmin: \the\textmin}%
1605 \fl@trace{\@spaces space on page = \the \@dbltoproom.
1606 Ht float: \the \ht \currbox-BANG}%
1607 </trace>
1608 \ifdim \@dbltoproom>\ht\currbox
1609 \tempswatru
1610 <*trace>
1611 \fl@trace{Space OK BANG: space on page =
1612 \the \@dbltoproom > \the \ht \currbox}%
1613 \else
1614 \fl@trace{fpstype: \the \fpstype}%
1615 \fl@trace{Fail---no room dbltoproom-BANG?:}%
1616 \fl@trace{\@spaces space on page = \the \@dbltoproom.
1617 Ht float: \the \ht \currbox}%
1618 </trace>
1619 \fi
1620 \advance \@dbltoproom -\textmin
1621 <*trace>
1622 \else
1623 \fl@trace{fpstype: \the \fpstype}%
1624 \fl@trace{Fail---no room dbltoproom-ORD?:}%
1625 \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1626 Ht float: \the \ht \currbox}%
1627 </trace>
1628 \fi
1629 \fi
1630 \if@tempswa
1631 \bitor \currtype \deferlist
1632 <*trace>
1633 \fl@trace{(dbl)deferlist: \deferlist: (before)}%
1634 </trace>
 not in fixfloats?
1635 \testwidth\currbox
1636 \if@test
1637 <*trace>
1638 \fl@trace{type already on list: (dbl)defer}%
1639 </trace>
1640 \else
1641 \tempdima -\ht\currbox

```

```

1642 \advance\@tempdima
1643 -\ifx \dbltoplist\empty \dbltxtfloatsep \else
1644 \dblfloatsep \fi
1645 \global \advance \dbltoproom \@tempdima
1646 \global \advance \colht \@tempdima
1647 \global \advance \dbltopnum \m@ne
1648 \cons \dbltoplist \currbox
1649 <*trace>
1650 \f@trace{dbltopnum (after) = \the \dbltopnum}%
1651 \f@trace{***Success: dbltop}%
1652 </trace>
1653 \cinserttrue
1654 \fi
1655 \fi
1656 <*trace>
1657 \else
1658 \f@trace{Fail: dbltopnum = \the \dbltopnum: fpstype
1659 \the \fpstype=ORD?}%
1660 \ifnum \fpstype<\sixt@n
1661 \f@trace{ERROR: !t float not successful (addtoblcol)}%
1662 \fi
1663 </trace>
1664 \fi
1665 \fi
1666 \cinsert
1667 \else
1668 <*trace>
1669 \f@trace{put on deferlist}%
1670 </trace>
1671 \cons\@deferlist\currbox
1672 <*trace>
1673 \f@trace{(dbl)deferlist: \@deferlist: (after)}%
1674 </trace>
1675 \fi
1676 <*trace>
1677 \f@trace{End of addtoblcol -- locally count:}%
1678 \f@trace{ dbltop: \the \dbltopnum.}%
1679 </trace>
1680 \endgroup
1681 <*trace>
1682 \f@trace{End of addtoblcol -- globally count:}%
1683 \f@trace{dbltop: \the \dbltopnum.}%
1684 </trace>
1685 }%
1686 </2ekernel | latexrelease | fltrace>
1687 <(latexrelease | fltrace)\EndIncludeInRelease
1688 <(latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
1689 <(latexrelease | fltrace) { \addtoblcol{float order in 2-column}}%
1690 <(latexrelease | fltrace)\def\addtoblcol{%
1691 <(latexrelease | fltrace) \begingroup
1692 <*trace>
1693 <(latexrelease | fltrace) \f@trace{***Start addtoblcol}%
1694 </trace>
1695 <(latexrelease | fltrace) \cinsertfalse

```

```

1696 〈latexrelease | fltrace〉 \csetfloattypecounts
1697 〈latexrelease | fltrace〉 \cgetfpsbit \tw@
1698 〈*trace〉
1699 〈latexrelease | fltrace〉 \fl@trace{fpstype \ifodd \ctempcnta OK
1700 〈latexrelease | fltrace〉 \else not \fi dbltop: \the \cfpstype}%
1701 〈/trace〉
1702 〈latexrelease | fltrace〉 \ifodd\ctempcnta
1703 〈latexrelease | fltrace〉 \cflsetnum \cdbltopnum
1704 〈latexrelease | fltrace〉 \ifnum \cdbltopnum>\z@
1705 〈latexrelease | fltrace〉 \ctempswafalse
1706 〈latexrelease | fltrace〉 \ifdim \cdbltoproom>\ht\currbox
1707 〈latexrelease | fltrace〉 \ctempswatrue
1708 〈*trace〉
1709 〈latexrelease | fltrace〉 \fl@trace{Space OK: \cdbltoproom =
1710 〈latexrelease | fltrace〉 \the \cdbltoproom > \the \ht \currbox
1711 〈latexrelease | fltrace〉 (dbltoproom)}%
1712 〈/trace〉
1713 〈latexrelease | fltrace〉 \else
1714 〈*trace〉
1715 〈latexrelease | fltrace〉 \fl@trace{fpstype: \the \cfpstype (addtoblc)}%
1716 〈/trace〉
1717 〈latexrelease | fltrace〉 \ifnum \cfpstype<\sixt@n
1718 〈*trace〉
1719 〈latexrelease | fltrace〉 \fl@trace{BANG float ignoring \cdbltoproom}%
1720 〈latexrelease | fltrace〉 \fl@trace{\cspaces \cdbltoproom =
1721 〈latexrelease | fltrace〉 \the \cdbltoproom.
1722 〈latexrelease | fltrace〉 Ht float: \the \ht \currbox-BANG}%
1723 〈/trace〉

```

Need to check that there is room on the page, using the local value of \textmin to make the necessary adjustment to \cdbltoproom.

```

1724 〈latexrelease | fltrace〉 \advance \cdbltoproom \textmin
1725 〈*trace〉
1726 〈latexrelease | fltrace〉 \fl@trace{Local value of texmin: \the\textmin}%
1727 〈latexrelease | fltrace〉 \fl@trace{\cspaces space on page =
1728 〈latexrelease | fltrace〉 \the \cdbltoproom.
1729 〈latexrelease | fltrace〉 Ht float: \the \ht \currbox-BANG}%
1730 〈/trace〉
1731 〈latexrelease | fltrace〉 \ifdim \cdbltoproom>\ht\currbox
1732 〈latexrelease | fltrace〉 \ctempswatrue
1733 〈*trace〉
1734 〈latexrelease | fltrace〉 \fl@trace{Space OK BANG: space on page =
1735 〈latexrelease | fltrace〉 \the\cdbltoproom > \the\ht\currbox}%
1736 〈latexrelease | fltrace〉 \else
1737 〈latexrelease | fltrace〉 \fl@trace{fpstype: \the \cfpstype}%
1738 〈latexrelease | fltrace〉 \fl@trace{Fail---no room dbltoproom-BANG?:}%
1739 〈latexrelease | fltrace〉 \fl@trace{\cspaces space on page =
1740 〈latexrelease | fltrace〉 \the \cdbltoproom.
1741 〈latexrelease | fltrace〉 Ht float: \the \ht \currbox}%
1742 〈/trace〉
1743 〈latexrelease | fltrace〉 \fi
1744 〈latexrelease | fltrace〉 \advance \cdbltoproom -\textmin
1745 〈*trace〉
1746 〈latexrelease | fltrace〉 \else

```

```

1747 <{latexrelease | fltrace>
1748 <{latexrelease | fltrace>
1749 <{latexrelease | fltrace>
1750 <{latexrelease | fltrace>
1751 <{latexrelease | fltrace>
1752 </trace>
1753 <{latexrelease | fltrace>
1754 <{latexrelease | fltrace>
1755 <{latexrelease | fltrace>
1756 <{latexrelease | fltrace>
1757 <*trace>
1758 <{latexrelease | fltrace>
1759 <{latexrelease | fltrace>
1760 </trace>
1761 <{latexrelease | fltrace>
1762 <*trace>
1763 <{latexrelease | fltrace>
1764 </trace>
1765 <{latexrelease | fltrace>
1766 <{latexrelease | fltrace>
1767 <{latexrelease | fltrace>
1768 <{latexrelease | fltrace>
1769 <{latexrelease | fltrace>
1770 <{latexrelease | fltrace>
1771 <{latexrelease | fltrace>
1772 <{latexrelease | fltrace>
1773 <{latexrelease | fltrace>
1774 <{latexrelease | fltrace>
1775 <*trace>
1776 <{latexrelease | fltrace>
1777 <{latexrelease | fltrace>
1778 <{latexrelease | fltrace>
1779 </trace>
1780 <{latexrelease | fltrace>
1781 <{latexrelease | fltrace>
1782 <{latexrelease | fltrace>
1783 <*trace>
1784 <{latexrelease | fltrace>
1785 <{latexrelease | fltrace>
1786 <{latexrelease | fltrace>
1787 <{latexrelease | fltrace>
1788 <{latexrelease | fltrace>
1789 <{latexrelease | fltrace>
1790 <{latexrelease | fltrace>
1791 </trace>
1792 <{latexrelease | fltrace>
1793 <{latexrelease | fltrace>
1794 <{latexrelease | fltrace>
1795 <{latexrelease | fltrace>
1796 <*trace>
1797 <{latexrelease | fltrace>
1798 </trace>
1799 <{latexrelease | fltrace>
1800 <*trace>
```

\fl@trace{fpstype: \the \@fpstype}%
\fl@trace{Fail---no room dbltoproom-ORD?:}%
\fl@trace{\@spaces \@dbltoproom =
\the \@dbltoproom.
Ht float: \the \ht \@currbox}%
\fi
\fi
\if@tempswa
\obitor \currtype \@dbldeferlist
\fl@trace{dbldeferlist:
\@dbldeferlist: (before)}%
\if@test
\fl@trace{type already on list: dbldefer}%
\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
\fl@trace{dbltopnum (after) =
\the \@dbltopnum}%
\fl@trace{\*\*\*Success: dbltop}%
\@inserttrue
\fi
\fi
\else
\fl@trace{Fail: dbltopnum = \the \@dbltopnum:
fpstype \the \@fpstype=ORD?}%
\ifnum \@fpstype<\sixt@n
\fl@trace{ERROR: !t float not successful
(addtodbcol)}%
\fi
\fi
\if@insert
\else
\fl@trace{put on dbldeferlist}%
\@cons\@dbldeferlist\@currbox
\fi

```

1801 <{latexrelease | fltrace} \f1@trace{dbldeferlist: \@dbldeferlist: (after)}%
1802 </trace>
1803 <{latexrelease | fltrace} \fi
1804 <{*trace}>
1805 <{latexrelease | fltrace} \f1@trace{End of addtodblcol -- locally count:}%
1806 <{latexrelease | fltrace} \f1@trace{ dbltop: \the \@dbltopnum.}%
1807 </trace>
1808 <{latexrelease | fltrace} \endgroup
1809 <{*trace}>
1810 <{latexrelease | fltrace} \f1@trace{End of addtodblcol -- globally count:}%
1811 <{latexrelease | fltrace} \f1@trace{dbltop: \the \@dbltopnum.}%
1812 </trace>
1813 <{latexrelease | fltrace}>%
1814 <{latexrelease | fltrace}\EndIncludeInRelease

```

(End of definition for \@addtodbcol.)

### \@addmarginpar

```

1815 <{*2ekernel}>
1816 \def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
1817 \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
1818 \if@twocolumn
1819 \if@firstcolumn \@tempcnta\m@ne \fi
1820 \else
1821 \if@mparswitch
1822 \ifodd\c@page \else\@tempcnta\m@ne \fi
1823 \fi
1824 \if@reversemargin \@tempcnta -\@tempcnta \fi
1825 \fi
1826 \ifnum\@tempcnta <\z@ \global\setbox\@marbox\box\@currbox \fi
1827 \@tempdima\@mparbottom
1828 \advance\@tempdima -\@pageht
1829 \advance\@tempdima\ht\@marbox
1830 \ifdim\@tempdima >\z@
1831 \@latex@warning@no@line {Marginpar on page \thepage\space moved}%
1832 \else
1833 \@tempdima\z@
1834 \fi
1835 \global\@mparbottom\@pageht
1836 \global\advance\@mparbottom\@tempdima
1837 \global\advance\@mparbottom\dp\@marbox
1838 \global\advance\@mparbottom\marginparpush
1839 \advance\@tempdima -\ht\@marbox

```

Putting box movement inside the ‘marbox’:

```

1840 \global\setbox \@marbox
1841 \vbox {\vskip \@tempdima
1842 \box \@marbox}%
1843 \global \ht\@marbox \z@
1844 \global \dp\@marbox \z@

```

Sticking (rather than gluing:-) the ‘marbox’ to the line above, changed vskip to kern:

```

1845 \kern -\@pagedp
1846 \nointerlineskip
1847 \hb@xt@\columnwidth

```

```

1848 {\ifnum \c@tempcnta > \z@%
1849 \hskip\columnwidth \hskip\marginparsesep%
1850 \else%
1851 \hskip -\marginparsesep \hskip -\marginparwidth%
1852 \fi%
1853 \box\c@marbox \hss}%

```

For this reason the following code can vanish:

```

\nobreak %% No longer needed. CAR92/12
\vskip -\c@tempdima %% No longer needed. CAR92/12

1854 \nointerlineskip
1855 \hbox{\vrule \c@height\z@ \c@width\z@ \c@depth\c@pagedp}%

```

*(End of definition for \c@addmarginpar.)*

### 1.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for L<sup>A</sup>T<sub>E</sub>X2e.

\enlargethispage{<dim>}

Adds <dim> to the height of the current column only. On the printed page the bottom of this column is extended downwards by exactly <dim> without having any effect on the placement of the footer; this may result in an overprinting.

\enlargethispage\*{<dim>}

Similar to \enlargethispage but it tries to squeeze the column to be printed in as small a space as possible, ie it uses any shrinkability in the column. If the column was not explicitly broken (e.g. with \pagebreak) this may result in an overfull box message but except for this it will come out as expected (if you know what to expect).

The star form of this command is dedicated to Leslie Lamport, the other we need for ourselves (FMi, CAR).

These commands may well have unwanted effects if used soon before a \clearpage: please give keep them clear of such places.

\@kludgeins The insert which makes T<sub>E</sub>X do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```

1856 \newinsert \@kludgeins
1857 \global\dimen@\@kludgeins \maxdimen
1858 \global\count@\@kludgeins 1000

```

*(End of definition for \@kludgeins.)*

\enlargethispage The user command.

```

1859 \gdef \enlargethispage {%
1860 \c@ifstar
1861 {%
1862 \c

```

```

1866 {%
1867 <*trace>
1868 \f@trace{Enlarging page height exactly---}%
1869 </trace>
1870 \enlargepage\empty}%
1871 }

```

(End of definition for `\enlargepage` and `\enlargepage*`.)

- `\enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```

1872 \gdef\enlargepage#1#2{%
1873 <*trace>
1874 \f@trace{\@spaces\@spaces by #2}%
1875 </trace>
1876 \@tempskipa#2\relax
1877 \ifdim \@tempskipa>.5\maxdimen
1878 \@latex@error{Suggested\space extra\space height\space
1879 (the\@tempskipa)\space dangerously\space
1880 large}\@eha
1881 \else
1882 \ifdim \vsize<.5\maxdimen
1883 <*trace>
1884 \f@trace {Kludgeins added--pagegoal before: \the\pagegoal}%
1885 </trace>
1886 \@bsphack
1887 \insert\kludgeins{#1\vskip-\@tempskipa}%
1888 \@esphack

```

This next bit is for tracing only:

```

1889 <*trace>
1890 \ifvmode \par
1891 \f@trace {Kludgeins added--pagegoal after: \the \pagegoal}%
1892 \fi
1893 </trace>
1894 \else
1895 \@latex@error{Page\space height\space already\space
1896 too\space large}\@eha
1897 \fi
1898 \fi
1899 }

```

(End of definition for `\enlargepage`.)

- `\ShowFloat` This command provides some information about the contents of a float register. Float registers have internal names of the form `\bx@Uppercase-letter(s)-or numbers` and you specify just this letter or letters as the argument, e.g., `\ShowFloat{A}`. (There is not much error recovery if you specify something that isn't a float.)

```

1900 </2ekernel>
1901 <*2ekernel | latexrelease>
1902 <latexrelease>\IncludeInRelease{2021/11/15}%
1903 <latexrelease> \ShowFloat{Show float register contents}%
1904 \def\ShowFloat#1{\begingroup
1905 \let \f@trace \f@tracemessage
1906 \f@trace{***Float #1 details:}%
1907 \ifcsname bx@\#1\endcsname

```

```

1908 \expandafter\fl@ShowFloat\csname bx@\#1\endcsname
1909 \else
1910 \fl@trace{Not a float!}%
1911 \fi
1912 \endgroup
1913 }
1914 \def\fl@ShowFloat#1{%
1915 \fl@traceval{\count#1}%
1916 % this here should be interpreted on day
1917 \fl@traceval{\ht#1}%
1918 \fl@traceval{\dp#1}%
1919 \fl@traceval{\wd#1}%
1920 {\tracingonline1\showboxbreadth10\showboxdepth3\showbox#1}%
1921 }
```

Here are two definitions from `fltrace` that make the above code work:

```

1921 \def \fl@traceval #1{\fl@trace{\string #1 = \the #1}}
1922 \def \fl@tracemessage #1{{\let\@elt\@empty\typeout{LaTeX2e: #1}}}
1923 {/2ekernel | latexrelease}
1924 {/latexrelease}\EndIncludeInRelease

1925 {/latexrelease}\IncludeInRelease{0000/00/00}%
1926 {/latexrelease} {\ShowFloat}{Show float register contents}%
1927 {/latexrelease}
1928 {/latexrelease}\let\ShowFloat\@undefined
1929 {/latexrelease}\let\fl@ShowFloat\@undefined
1930 {/latexrelease}\let\fl@traceval\@undefined
1931 {/latexrelease}\let\fl@tracemessage\@undefined
1932 {/latexrelease}\EndIncludeInRelease
```

*(End of definition for `\ShowFloat`.)*

### 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 following 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.

```
\f@trace \tracefloatoff Set-up tracing for floats independent of other tracing as it produces mega-output. Default
\tracefloats is no tracing.
1933 {*f@trace}
1934 \def \f@tracemessage #1{\let\@elt\empty\typeout{LaTeX2e: #1}}
1935 \def \tracefloats{\let \f@trace \f@tracemessage}
1936 \def \tracefloatoff {\let \f@trace \@gobble}
1937 \tracefloatoff
1938 \def \f@traceval #1{\f@trace{\string #1 = \the #1}}
1939 \IncludeInRelease{2015/01/01}{\tracefloatvals}%
1940 {trace float vals}%
1941 \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.<sup>60</sup>

```
1942 \begingroup
```

When the user requests \tracefloatvals then they should show regardless of the tracing state, so locally we make sure that it is activated.

```
1943 \tracefloats
1944 \dblfloatplacement
1945 \floatplacement
1946 \f@trace{***Float placement parameters:}%
1947 \f@traceval\@colnum
1948 \f@traceval\@colroom
1949 \f@traceval\@topnum
1950 \f@traceval\@toproom
1951 \f@traceval\@botnum
1952 \f@traceval\@botroom
1953 \f@traceval\@fpmin
1954 \f@trace{\string\textration = \textration}%
1955 \f@traceval\@dbltopnum
1956 \f@traceval\@dbltoproom
1957 \f@trace{\string\textration = \textration}%
1958 \f@trace{toplist: \@toplist}%
1959 \f@trace{botlist: \@botlist}%
1960 \f@trace{midlist: \@midlist}%
1961 \f@trace{deferlist: \@deferlist}%
1962 \f@trace{dbltoplist: \@dbltoplist}%
1963 %FMi \f@trace{dbldeferlist: \@dbldeferlist}%
1964 \endgroup
1965 }
1966 \EndIncludeInRelease
1967 \IncludeInRelease{0000/00/00}{\tracefloatvals}%
1968 {trace float vals}%
1969 \def \tracefloatvals{%
```

---

<sup>60</sup>This is a somewhat questionable design.

```

1970 \begingroup
1971 \tracefloats
1972 \@dblfloatplacement
1973 \@floatplacement
1974 \fl@trace{***Float placement parameters:}%
1975 \fl@traceval\@colnum
1976 \fl@traceval\@colroom
1977 \fl@traceval\@topnum
1978 \fl@traceval\@toproom
1979 \fl@traceval\@botnum
1980 \fl@traceval\@botroom
1981 \fl@traceval\@fpmin
1982 \fl@trace{\string\textration = \textfraction}%
1983 \fl@traceval\@dbltopnum
1984 \fl@traceval\@dbltoproom
1985 \fl@trace{\string\textration = \textfraction}%
1986 \fl@trace{toplist: \@toplist}%
1987 \fl@trace{botlist: \@botlist}%
1988 \fl@trace{midlist: \@midlist}%
1989 \fl@trace{deferlist: \@deferlist}%
1990 \fl@trace{dbltoplist: \@dbltoplist}%
1991 % next line only in old releases
1992 \fl@trace{dbldeferlist: \@dbldeferlist}%
1993 \endgroup
1994 }
1995 \EndIncludeInRelease

```

We need to make sure that `fltrace` comes before `flafter` to make the tracing work.

```

1996 \Qifpackageloaded{flafter}
1997 { \PackageWarningNoLine
1998 {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
1999 Attempting to recover by reloading 'flafter')}%

```

Hide the fact that `flafter` was already loaded and then request it anew.

```

2000 \expandafter\let\csname ver@flafter.sty\endcsname\relax
2001 \def\reserved@a{\relax\@gobble}
2002 \expandafter\let\csname string#1+flafter+IIR\endcsname\relax}%
2003 \reserved@a\@addtocurcol
2004 \reserved@a\@addtonextcol
2005 \RequirePackage{flafter}{}}
2006
```

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.

```

2007 (*flafter)
2008 \providecommand\fl@trace[1]{}
2009
```

*(End of definition for `\fl@trace` and others.)*

`\suppressfloats` Float suppression commands: these set the relevant counter globally to zero. Thus they are overridden for a particular float by an `!` specifier.

```

2010 {*2ekernel}
2011 \def \suppressfloats {%
2012 \Qifnextchar [%

```

```

2013 \@flstop
2014 {\global \@colnum \z@}%
2015 }

```

Maybe this should be a loop over #1?

```

2016 \def \@flstop [#1]{%
2017 \if t#1%
2018 \global \@topnum \z@
2019 \fi
2020 \if b#1%
2021 \global \@botnum \z@
2022 \fi
2023 }

```

*(End of definition for \suppressfloats and \@flstop.)*

Manipulation of float placement and type; both their strings and the corresponding count registers.

\@fpstype First a new count register to go with \currtype.

\@reqcolroom Then a new skip register, for information needed to remove the \maxsep conservatism: it is possible that this could use a temporary register.

\@textfloatsheight 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.

```

2024 \newcount \@fpstype
2025 \newdimen \@reqcolroom
2026 \newdimen \@textfloatsheight
2027
```

*(End of definition for \@fpstype, \@reqcolroom, and \@textfloatsheight.)*

\@fpsadddefault Adds the default placement to what is already there.

Should not need to change this, but could do it as follows:

```

def \@fpsadddefault {%
 @temptokena \expandafter\expandafter\expandafter
 {\csname fps@\@capttype \endcsname}%
 \edef \reserved@a {\the\@temptokena}%
 @onelevel@sanitize \reserved@a
 \edef \@fps {\@fps\reserved@a}%

2028 <*2ekernel | fltrace>
2029 \def \@fpsadddefault {%
2030 <*trace>
2031 \f@trace{fps changed from: \@fps}%
2032 }</trace>
2033 \edef \@fps {\@fps\csname fps@\@capttype \endcsname}%
2034 @latex@warning {%
2035 No positions in optional float specifier.\MessageBreak
2036 Default added (so using '\@fps')}%
2037 }

```

*(End of definition for \@fpsadddefault.)*

\@setfloattypecounts Sets counters \@fpstype and \currtype.  
BANG == bit4 of \count\@currbox = 0.

```

2038 \def \@setfloattypecounts {%
2039 \currtype \count\@currbox
2040 \@fpstype \count\@currbox
2041 \divide\currtype\@xxxii \multiply\currtype\@xxxii
2042 \advance \@fpstype -\currtype
2043 (*trace)
2044 \f@trace{(mod 32) fpstype: \the \@fpstype}%
2045 \f@trace{((mult of 32) currtype: \the \@currtype)}%
2046 % Tracing only: but some should be changed into real errors/warnings?
2047 \ifnum \@fpstype<\sixt@n
2048 \ifnum \@fpstype=\z@
2049 \f@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
2050 \fi
2051 \ifnum \@fpstype=\ne
2052 \f@trace{WARNING: only h, fpstype = \the \@fpstype = 1?}%
2053 \fi
2054 \f@trace{BANG float}%
2055 \else
2056 \ifnum \@fpstype=\sixt@n
2057 \f@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
2058 \fi
2059 \ifnum \@fpstype=17
2060 \f@trace{WARNING: only h, fpstype = \the \@fpstype = 17?}%
2061 \fi
2062 \f@trace{ORD float}%
2063 \fi
2064
```

(End of definition for \@setfloattypecounts.)

Macros for getting, testing and setting bits of the fps.

\@getfpsbit Sets \@tempcnta to required bit of \count\@currbox.

```

2067 <*2ekernel>
2068 \def \@getfpsbit {%
2069 \@boxfpsbit \@currbox
2070 }
```

(End of definition for \@getfpsbit.)

\@boxfpsbit Used above.

```

2071 \def \@boxfpsbit #1#2{%
2072 \@tempcnta \count#1%
2073 \divide \@tempcnta #2\relax
2074 }
```

(End of definition for \@boxfpsbit.)

\@testfp New definition of the float page test.

```

2075 \def \@testfp #1{%
2076 \@boxfpsbit #18\relax % Really '#1 8' for human readers!
```

```

2077 \ifodd \@tempcnta
2078 \else
2079 \@testtrue
2080 \fi
2081 }
```

(End of definition for \testfp.)

\@setfpsbit Sets required bit of \@tempcnta (to 1).

```

2082 \def \@setfpsbit #1{%
2083 \@tempcntb \@tempcnta
2084 \divide \@tempcntb #1\relax
2085 \ifodd \@tempcntb
2086 \else
2087 \advance \@tempcnta #1\relax
2088 \fi
2089 }
2090 </2ekernel>
```

(End of definition for \setfpsbit.)

\@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.

```

2091 <*2ekernel | ftrace>
2092 \def \@resethfps {%
2093 \let\reserved@a\empty
2094 \ifnum \fpstype=\one
2095 \def \reserved@a {!}{%
2096 \fpstype 17
2097 \fi
2098 \ifnum \fpstype=17
2099 \global \advance \count\currbox \tw@
2100 \@latex@warning@no@line {%
2101 '\reserved@a h' float specifier changed to '\reserved@a ht'}%
2102 <*trace>
2103 \f@trace{%
2104 't' added to '\reserved@a h'- new Count: \the \count\currbox}%
2105 </trace>
2106 \fi
2107 }
```

(End of definition for \resethfps.)

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 `\@addtobdblcol`. There may be more efficient ways to handle this, but the group seems to be the simplest.

```

2108 \def \@flsetnum #1{%
2109 <*trace>
2110 \f1@trace{fpstype: \the \@fpstype (flsetnum \string#1)}%
2111
```

`\ifnum \@fpstype<\sixt@n`

`\ifnum #1=\z@`

`(*trace)`

`\f1@trace{BANG float resetting \string#1 to 1}%`

`(/trace)`

`#1\@ne`

`\fi`

`\fi`

`(*trace)`

`\f1@trace{\#1 (before) = \the \#1}%`

`(/trace)`

`}`

*(End of definition for `\@flsetnum`.)*

`\@flsettextmin` This ignores `\textfraction` space restriction in case BANG.

```

2124 \def \@flsettextmin {%
2125 <*trace>
2126 \f1@trace{fpstype: \the \@fpstype (flsettextmin)}%
2127
```

`\ifnum \@fpstype<\sixt@n`

`(*trace)`

`\f1@trace{BANG ignoring textmin}%`

`(/trace)`

`\@textmin \z@`

`\else`

`\@textmin \textfraction\@colht`

`(*trace)`

`\f1@trace{ORD textmin = \the \@textmin}%`

`(/trace)`

`\fi`

`}`

*(End of definition for `\@flsettextmin`.)*

`\@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`.

```

2140 \def \@flcheckspace #1#2{%
2141 \advance \reqcolroom
2142 \ifx #2\empty \textfloatsep \else \floatsep \fi
2143
```

`(*trace)`

`\f1@trace{colroom = \the \@colroom}`

`(flcheckspace \string#1 \string#2)%`

`\f1@trace{reqcolroom = \the \reqcolroom}`

`(flcheckspace \string#1 \string#2)%`

`(/trace)`

```

2149 \ifdim \@colroom>\@reqcolroom
2150 \ifdim #1>\ht\@currbox
2151 \tempswatru
2152 <*trace>
2153 \f@trace{Space OK: #1 = \the #1 > \the \ht \@currbox
2154 (flcheckspace \string#1 \string#2)}%
2155 </trace>
2156 \else
2157 <*trace>
2158 \f@trace{fpstype: \the \@fpstype
2159 (flcheckspace \string#1 \string#2)}%
2160 </trace>
2161 \ifnum \@fpstype<\sixt@n
2162 <*trace>
2163 \f@trace{BANG float ignoring #1
2164 (flcheckspace \string#1 \string#2):}%
2165 \f@trace{@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2166 BANG}%
2167 </trace>
2168 \tempswatru
2169 <*trace>
2170 \else
2171 \f@trace{Fail---no room (flcheckspace \string#1 \string#2)
2172 (fpstype \the \@fpstype=ORD?):}%
2173 \f@trace{@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2174 ORD?}%
2175 </trace>
2176 \fi
2177 \fi
2178 <*trace>
2179 \else
2180 \f@trace{Fail---no room at 2nd test of colroom
2181 (flcheckspace \string#1 \string#2)}%
2182 </trace>
2183 \fi
2184 }
2185 </2ekernel | fltrace>

```

(End of definition for \@flcheckspace.)

\@flupdates This updates everything when a float is placed.

```

2186 <*2ekernel>
2187 \def \@flupdates #1#2#3{%
2188 \global \advance #1\m@ne
2189 \global \advance \@colnum \m@ne
2190 \tempdima -\ht\@currbox
2191 \advance \tempdima
2192 -\ifx #3\empty \textfloatsep \else \floatsep \fi
2193 \global \advance #2\tempdima
2194 \global \advance \@colroom \tempdima
2195 \cons #3\@currbox
2196 }
2197 </2ekernel>

```

(End of definition for \oflupdates.)

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 footnotes come out on the correct page? Or should we go for as much compatibility as possible (but see next item)?

17. Should we continue to support (as much as possible) `\samepage`? Some of its intended functionality is now advertised as being provided by `\enlargethispage`. Use of either is likely to result in wrongly placed footnotes, marginals, etc. Which should have priority: obeying the pagination instructions, or correct placement of notes/marginalia?

18. Is the adjustment of space to cause shrinking in the kludge-\* case correct? Should it be limited to 0pt?

19. Is the setting of `\boxmaxdepth` in `makecol` and friends needed? It only has any effect if `\@textbottom` ends with a box or rule, in which case the vskip to allow for its depth should also be added. If it is kept, it should probably be the last thing in the box. It has now been removed.

It would perhaps be better to document that `\@textbottom` and `\@texttop` must have natural height 0pt.

20. I cannot see why the vskip adjustment for the depth is needed if `boxmaxdepth` is used to ensure that there is never a too deep box.

21. The value of `\boxmaxdepth` should be explicitly set whenever necessary: it is too risky to assume that it has any particular value. Care is needed in deciding what to set it to.

It is interesting to note that the value of `\boxmaxdepth` is unique in being read before the local settings for the box group are reset; all other parameter settings which affect the box construction use their values outside the box group.

22. Should `\@maxdepth` store the setting of `\maxdepth` from `lplain`? Or should we provide a proper interface to class files for setting these?

An analysis of various other macros.

`\@opcol` should do `\@floatplacement`, but where? Right at the end, since it always occurs at the start of a column.

```
\def\@opcol{%
 % Why is this done first?
 \global \c@parbottom \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.

2198 <{latexrelease | fltrace}>\IncludeInRelease{2017/01/01}%
2199 <{latexrelease | fltrace}> {\@makefcolumn}{negative height floats}%
2200 <{/2ekernel | fltrace | latexrelease}%
2201 \def\@makefcolumn #1{%
2202 \begingroup
2203 \fpmin -\maxdimen
2204 \let \testfp \gobble
2205 \tryfcolumn #1%
2206 \endgroup
2207 {*trace}
2208 \if@fcollmade
2209 \f@trace{PAGE: in \string\clearpage
2210 \if@twocolumn ---twocolumn\fi---}%
2211 \f@trace{---- float column/page completed from \string#1}%
2212 \fi
2213 /{trace}
2214 }

2215 <{latexrelease | fltrace}>\EndIncludeInRelease
2216 <{latexrelease | fltrace}>\IncludeInRelease{0000/00/00}%
2217 <{latexrelease | fltrace}> {\@makefcolumn}{negative height floats}%
2218 <{latexrelease | fltrace}>\def\@makefcolumn #1{%
2219 \begingroup
2220 \fpmin \z@
2221 \let \testfp \gobble
2222 \tryfcolumn #1%
2223 \endgroup
2224 {*trace}
2225 \if@fcollmade
2226 \f@trace{PAGE: in \string\clearpage
2227 \if@twocolumn ---twocolumn\fi---}%
2228 \f@trace{---- float column/page completed
2229 from \string#1}%
2230 \fi
2231 /{trace}
2232 \endgroup
2233 \EndIncludeInRelease
2234 <{/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.

```
2235 <{latexrelease | fltrace}>\IncludeInRelease{2015/01/01}%
```

```

2236 <{latexrelease | fltrace} {\\@outputdblcol}{2 column marks}%
2237 {*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.

```

2238 \\def\\@outputdblcol{%
2239 \\if@firstcolumn
2240 \\global\\@firstcolumnfalse

```

Save the left column

```

2241 \\global\\setbox\\@leftcolumn\\copy\\@outputbox
2242 <{fltrace} \\f@trace{PAGE: first column boxed}%

```

Remember the marks from the first column

```

2243 \\splitmaxdepth\\maxdimen
2244 \\vbadness\\maxdimen

```

In case of `\enlargeithispage` 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 `\vspli` 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.

```

2245 \\setbox\\@outputbox\\vbox{\\unvbox\\@outputbox\\unskip}%
2246 \\setbox\\@outputbox\\vspli\\@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`.

```

2247 \\toks@\\expandafter{\\topmark}%
2248 \\xdef\\@firstcoltopmark{\\the\\toks@}%
2249 \\toks@\\expandafter{\\splitfirstmark}%
2250 \\xdef\\@firstcolfirstmark{\\the\\toks@}%

```

This test does not work if truly empty marks have been inserted, but L<sup>A</sup>T<sub>E</sub>X 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.)

```

2251 \\ifx\\@firstcolfirstmark\\empty
2252 \\global\\let\\@setmarks\\relax
2253 \\else
2254 \\gdef\\@setmarks{%
2255 \\let\\firstmark\\@firstcolfirstmark
2256 \\let\\topmark\\@firstcoltopmark}%
2257 \\fi

```

End of change

```

2258 \\else
2259 \\global\\@firstcolumntrue
2260 \\setbox\\@outputbox\\vbox{%
2261 \\hb@xt@\\textwidth{%
2262 \\hb@xt@\\columnwidth{\\box\\@leftcolumn \\hss}}%
2263 \\hfil

```

The color of the `\vrule` should be `\normalcolor` as to not inherit the color from the column.

```

2264 {\\normalcolor\\vrule \\@width\\columnseprule}%
2265 \\hfil
2266 \\hb@xt@\\columnwidth{\\box\\@outputbox \\hss}}}%
2267 <{fltrace} \\f@trace{PAGE: second column also boxed}%
2268 \\@combinedblfloats

```

Override current first and top with those of first column if necessary

```
2269 \@setmarks
230 End of change
231
2320 \@outputpage
2321 <fltrace> \f@l@trace{PAGE: two column page completed}%
2322 \begingroup
2323 \@dblfloatplacement
2324 \@startdblcolumn
2325 \@whilesw\if@fcolmade \fi{\@outputpage
2326 <fltrace> \f@l@trace{PAGE: double float page completed}%
2327 \@startdblcolumn}%
2328 \endgroup
2329 \fi}%
2330
2331 <latexrelease | fltrace>\EndIncludeInRelease
2332 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
2333 <latexrelease | fltrace> {\@outputdblcol}{2 column marks}%
2334 <latexrelease | fltrace>\def\@outputdblcol{%
2335 <latexrelease | fltrace> \if@firstcolumn
2336 <latexrelease | fltrace> \global \@firstcolumnfalse
2337 <latexrelease | fltrace> \global \setbox\@leftcolumn \box\@outputbox
2338 <*trace>
2339 <latexrelease | fltrace> \f@l@trace{PAGE: first column boxed}%
2340 </trace>
2341 <latexrelease | fltrace> \else
2342 <latexrelease | fltrace> \global \@firstcolumntrue
2343 <latexrelease | fltrace> \setbox\@outputbox \vbox {%
2344 <latexrelease | fltrace> \hb@xt@\textwidth {%
2345 <latexrelease | fltrace> \hb@xt@\columnwidth {%
2346 <latexrelease | fltrace> \box\@leftcolumn \hss}%
2347 <latexrelease | fltrace> \hfil
2348 <latexrelease | fltrace> {\normalcolor\vrule
2349 <latexrelease | fltrace> \width\columnseprule}%
2350 <latexrelease | fltrace> \hfil
2351 <latexrelease | fltrace> \hb@xt@\columnwidth {%
2352 <latexrelease | fltrace> \box\@outputbox \hss}%
2353 <latexrelease | fltrace> }%
2354 <*trace>
2355 <latexrelease | fltrace> \f@l@trace{PAGE: second column also boxed}%
2356 </trace>
2357 <latexrelease | fltrace> \@combinedblfloats
2358 <latexrelease | fltrace> \@outputpage
2359 <*trace>
2360 <latexrelease | fltrace> \f@l@trace{PAGE: two column page completed}%
2361 </trace>
2362 <latexrelease | fltrace> \begingroup
2363 <latexrelease | fltrace> \@dblfloatplacement
2364 <latexrelease | fltrace> \@startdblcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startdblcolumn`.

```
2365 <latexrelease | fltrace> \@whilesw\if@fcolmade \fi
2366 <latexrelease | fltrace> {\@outputpage}
```

```

2317 {*trace}
2318 <| latexrelease | fltrace> \f1@trace{PAGE: double float page completed}%
2319 </| trace>
2320 <| latexrelease | fltrace> \@startdblcolumn}%
2321 <| latexrelease | fltrace> \endgroup
2322 <| latexrelease | fltrace> \fi
2323 <| latexrelease | fltrace>}%
2324 <| latexrelease | fltrace>\EndIncludeInRelease
2325 </| 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.
 

```

2326 {*2ekernel}
2327 \newcount\c@topnumber
2328 \setcounter{topnumber}{2}

```

(End of definition for \c@topnumber.)
- \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.
 

```

2329 \newcommand\topfraction{.7}

```

(End of definition for \topfraction.)
- \c@bottomnumber This counter holds the maximum number of floats that can appear at the bottom of a text page or column.
 

```

2330 \newcount\c@bottomnumber
2331 \setcounter{bottomnumber}{1}

```

(End of definition for \c@bottomnumber.)
- \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.
 

```

2332 \newcommand\bottomfraction{.3}

```

(End of definition for \bottomfraction.)
- \c@totalnumber This counter holds the maximum number of floats that can appear on any text page or column.
 

```

2333 \newcount\c@totalnumber
2334 \setcounter{totalnumber}{3}

```

(End of definition for \c@totalnumber.)

|                                    |                                                                                                                                                                                  |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\textfraction</code>         | This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text.                                                            |
|                                    | <small>2335 \newcommand\textfraction{.2}</small>                                                                                                                                 |
|                                    | <i>(End of definition for \textfraction.)</i>                                                                                                                                    |
| <code>\floatpagefraction</code>    | This macro holds the minimum proportion (as a decimal number) of a page or column that must be occupied by floating objects before a ‘float page’ is produced.                   |
|                                    | <small>2336 \newcommand\floatpagefraction{.5}</small>                                                                                                                            |
|                                    | <i>(End of definition for \floatpagefraction.)</i>                                                                                                                               |
| <code>\cdbltopnumber</code>        | This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page.                                                              |
|                                    | <small>2337 \newcount\cdbltopnumber</small>                                                                                                                                      |
|                                    | <small>2338 \setcounter{dbltopnumber}{2}</small>                                                                                                                                 |
|                                    | <i>(End of definition for \cdbltopnumber.)</i>                                                                                                                                   |
| <code>\dbltopfraction</code>       | This macro holds the maximum proportion (as a decimal number) of a two-column text page that can be occupied by double-column floats at the top.                                 |
|                                    | <small>2339 \newcommand\dbltopfraction{.7}</small>                                                                                                                               |
|                                    | <i>(End of definition for \dbltopfraction.)</i>                                                                                                                                  |
| <code>\dblfloatpagefraction</code> | This macro holds the minimum proportion (as a decimal number) of a page that must be occupied by double-column floating objects before a ‘double-column float page’ is produced. |
|                                    | <small>2340 \newcommand\dblfloatpagefraction{.5}</small>                                                                                                                         |
|                                    | <i>(End of definition for \dblfloatpagefraction.)</i>                                                                                                                            |

### Floats on a text page

`\floatsep`    When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters are used for both one-column mode and single-column floats in two-column mode. They are all rubber lengths.

`\floatsep` is the space between adjacent floats that are placed at the top or bottom of the text page or column.

`\textfloatsep` is the space between the main text and floats at the top or bottom of the page or column.

`\intextsep` is the space between in-text floats and the text.

```
2341 \newskip\floatsep
2342 \newskip\textfloatsep
2343 \newskip\intextsep
2344 \setlength\floatsep {12\p@ \oplus 2\p@ \ominus 2\p@}
2345 \setlength\textfloatsep{20\p@ \oplus 2\p@ \ominus 4\p@}
2346 \setlength\intextsep {12\p@ \oplus 2\p@ \ominus 2\p@}
```

*(End of definition for \floatsep, \textfloatsep, and \intextsep.)*

\dblfloatsep When double-column floats (floating objects that span the whole \textwidth) are placed at the top of a text page in two-column mode, the separation between the float and the text is controlled by \dblfloatsep and \dbltextfloatsep. They are rubber lengths.

\dblfloatsep is the space between adjacent double-column floats placed at the top of the text page.

\dbltextfloatsep is the space between the main text and double-column floats at the top of the page.

```
2347 \newskip\dblfloatsep
2348 \newskip\dbltextfloatsep
2349 \setlength\dblfloatsep {12\p@ \oplus 2\p@ \ominus 2\p@}
2350 \setlength\dbltextfloatsep{20\p@ \oplus 2\p@ \ominus 4\p@}
```

(End of definition for \dblfloatsep and \dbltextfloatsep.)

### Floats on their own page or column

\@fptop When floating objects are placed on a separate page or column, called a ‘float page’, the layout of the page is controlled by these parameters, which are rubber lengths.

\@fpsep At the top of the page \@fptop is inserted; typically this supplies some stretchable whitespace. At the bottom of the page \@fpbot is inserted. Between adjacent floats \@fpsep is inserted.

These parameters are used for all floating objects on a ‘float page’ in one-column mode, and for single-column floats in two-column mode.

Note that at least one of the two parameters \@fptop and \@fpbot should contain a plus ...fil so as to fill the remaining empty space.

```
2351 \newskip\@fptop
2352 \newskip\@fpsep
2353 \newskip\@fpbot
2354 \setlength\@fptop{0\p@ \oplus 1fil}
2355 \setlength\@fpsep{8\p@ \oplus 2fil}
2356 \setlength\@fpbot{0\p@ \oplus 1fil}
```

(End of definition for \@fptop, \@fpsep, and \@fpbot.)

\@dblfpsep Double-column ‘float pages’ in two-column mode use similar parameters.

```
2357 \newskip\@dblfpsep
2358 \newskip\@dblfpsep
2359 \newskip\@dblfpbot
2360 \setlength\@dblfpsep{0\p@ \oplus 1fill}
2361 \setlength\@dblfpbot{8\p@ \oplus 2fill}
2362 \setlength\@dblfpbot{0\p@ \oplus 1fill}
```

(End of definition for \@dblfpsep, \@dblfpbot, and \@dblfpbot.)

\topfigrule The macros can be used to put in rules between floats and text; whatever they insert \botfigrule should be vertical mode material which takes up zero space.

```
2363 \let\topfigrule=\relax
2364 \let\botfigrule=\relax
2365 \let\dblfigrule=\relax
2366 {/2ekernel}
```

(End of definition for \topfigrule, \botfigrule, and \dblfigrule.)

# File 55

## lttagging.dtx

```
1 {*2ekernel | latexrelease}
2 \ExplSyntaxOn
```

### 1 General support for tagged output

\SuspendTagging

\ResumeTagging The are places in code where it is import top stop any tagging activities, e.g., when we \tag\_suspend:n are doing trial typesetting that it is done several times. In such a case one must tag only \tag\_resume:n the final version that is actually used, otherwise tagging structures are generated which then do not end up in the PDF and confuse the mechanism. For this we have two commands that can be used in packages: \SuspendTagging and \ResumeTagging (with corresponding L3 programming layer commands). They are available as part of the L<sup>A</sup>T<sub>E</sub>X kernel, so that they can be safely used in packages whether or not tagging is requested. They all take a string argument that is used for debugging to easily identify why tagging was suspended or restarted, for example, in tabularx you find \SuspendTagging{tabularx}. By default these four commands do nothing.

The argument is used literally (in \typeout messages) without any expansion when debugging is turned on and otherwise it is not used at all. This means it is safe to write something like \SuspendTagging{\foo} or even \SuspendTagging\foo which means L<sup>A</sup>T<sub>E</sub>X has to parse only a single token instead of putting a string of characters into the argument. This means a tiny speed improvement but with many such debugging strings...

\UseTaggingSocket

Given that we sometimes have to suspend tagging, it would be fairly inefficient \tag\_socket\_use:n to put different plugs into these sockets whenever that happens. We therefore offer \tag\_socket\_use:nn \UseTaggingSocket which is like \UseSocket except that it expects a socket starting \tag\_socket\_use:nnn with tagsupport/ but the socket name is specified without this prefix, i.e.,

```
\UseTaggingSocket{foo} → \UseSocket{tagsupport/foo}
```

Beside being slightly shorter, the big advantage is that this way we can change \UseTaggingSocket to do nothing by switching a boolean instead of changing the plugs of the tagging support sockets back and forth.

Usually, these sockets have (beside the default plug defined for every socket) one additional plug defined and directly assigned. This plug is used when tagging is active. There may be more plugs, e.g., tagging with special debugging or special behaviour depending on the class or PDF version etc., but right now it is usually just on or off.

When tagging is suspended they all have the same predefined behaviour: The sockets with zero arguments do nothing. The sockets with one argument gobble their argument. The sockets with two arguments will drop their first argument and pass the second unchanged.

It is possible to use the tagging support sockets with \UseSocket directly, but in this case the socket remains active if \SuspendTagging is in force. There may be reasons for doing that but in general we expect to always use \UseTaggingSocket.

\UseExpandableTaggingSocket

For special cases like in some \halign contexts we need a fully expandable version \tag\_socket\_use\_expandable:n of the command. For these cases, \UseExpandableTaggingSocket can be used. To allow

being expandable, it does not output any debugging information if `\DebugSocketsOn` is in effect and therefore should be avoided whenever possible.

The L3 programming layer versions `\tag_socket_use_expandable:n`, `\tag_socket_use:n`, `\tag_socket_use:nn`, and `\tag_socket_use:nnn` are slightly more efficient than `\UseTaggingSocket` because they do not have to determine how many arguments the socket takes when disabling it.

## 2 Implementation

`\tag_suspend:n`      In the kernel, these commands get dummy definitions so that they can be used without harm in packages. The real definition is used when tagging gets enabled.

```

\SuspendTagging
\ResumeTagging
 3 \cs_new_eq:NN \tag_suspend:n \use_none:n
 4 \cs_new_eq:NN \tag_resume:n \use_none:n
 5 \cs_new_protected:Npn \SuspendTagging #1 { \tag_suspend:n {#1} }
 6 \cs_new_protected:Npn \ResumeTagging #1 { \tag_resume:n {#1} }
```

(End of definition for `\tag_suspend:n` and others.)

`\tag_socket_use:n`      Again this is not the final definition for the kernel; it is just a version to get going while some parts of the kernel support are still missing.

```

\tag_socket_use:nn
\tag_socket_use:nnn
\tag_socket_use_expandable:n
 7 \AddToHook{begindocument}[kernel]{
 8 \cs_if_exist:NF \tag_if_active:T
 9 {
10 \prg_new_conditional:Npnn \tag_if_active: { p , T , TF, F }
11 { \prg_return_false: }
12 }
13 }
```

Dummy definitions in the kernel. These definitions will get updated in `tagpdf`. The default in the kernel is just to get rid of the first argument, the second is preserved if present:

```

14 \cs_new:Npn \tag_socket_use_expandable:n #1 { }
15 \cs_new_protected:Npn \tag_socket_use:n #1 { }
16 \cs_new_protected:Npn \tag_socket_use:nn #1#2 { }
17 \cs_new_protected:Npn \tag_socket_use:nnn #1#2#3 { #3 }
18 \cs_new_protected:Npn \UseTaggingSocket #1 {
19 \int_case:nnF
20 { \int_use:c { c__socket_tagsupport/#1_args_int } }
21 {
22 0 \prg_do_nothing:
23 1 \use_none:n
24 2 \use_i:nn
```

We do not expect tagging sockets with more than one or two arguments, so for now we only provide those.

```

25 }
26 \ERRORusetaggingsocket % that should get a proper error message
27 }
```

The same as an expandable command:

```

28 \cs_new:Npn \UseExpandableTaggingSocket #1 {
29 \int_case:nnF
30 { \int_use:c { c__socket_tagsupport/#1_args_int } }
```

```

31 {
32 0 \prg_do_nothing:
33 1 \use_none:n
34 2 \use_i:nn
35 }
36 \ERRORuse taggingsocket % that should get a proper error message
37 }

```

*(End of definition for \tag\_socket\_use:n and others.)*

## 2.1 Tagging sockets

This collects tagging sockets that should be generally available so that they can also be used even if the tagging code is not loaded.

### 2.1.1 Tagging support for paragraph setup

Paragraphs are tagged through the code in the para/hooks. This code is sometimes adjusted, e.g. to produce a “flattened” paragraph or to use a different tag. Sockets related to such code parts are collected here.

\l\_\_tag\_block\_flattened\_level\_int

The block code needs to know if they are nested blockenvs inside a flattened environment. For this it uses a counter. Inside some contexts, e.g. at the begin of a minipage or a footnote this counter must be reset. We therefore define the counter here so that we can use it in the following socket.

```
38 \int_new:N \l__tag_block_flattened_level_int
```

*(End of definition for \l\_\_tag\_block\_flattened\_level\_int.)*

tagsupport/para/restore (*socket*) This socket restores the para related settings to their default. It should be used in places where “normal” paragraph tagging must be ensured, for example at the begin of a footnote.

```
39 \NewSocket{tagsupport/para/restore}{0}
```

default (*plug*)

```

40 \NewSocketPlug{tagsupport/para/restore}{default}
41 {
42 \tl_set:Nn \l__tag_para_main_tag_tl {text-unit}
43 \tl_set_eq:NN \l__tag_para_tag_tl\l__tag_para_tag_default_tl
44 \bool_set_false:N\l__tag_para_flattened_bool
45 \int_zero:N \l__tag_block_flattened_level_int
46 \bool_set_true:N \l__tag_para_bool
47 }
48 \AssignSocketPlug{tagsupport/para/restore}{default}
```

tagsupport/para/begin (*socket*) These sockets are currently defined in tagpdf. They overwrite definitions in the latex-lab- tagsupport/para/end (*socket*) block code. There is also a simpler definition that probably should be a general socket too. TODO: move this into lttagging.

### 2.1.2 Tagging socket for targets

`\tagsupport/refstepcounter (socket)` When tagging is active we want to track the current structure number when targets are set. This will be mostly used in `\refstepcounter` but also if targets are set manually.

```
49 \NewSocket{tagsupport/recordtarget}{0}
```

`\tagsupport/recordtarget) (plug)`

```
50 %
51 \NewSocketPlug{tagsupport/recordtarget}{kernel}
52 {
53 \tl_if_blank:VF \@currentHref
54 {
55 \prop_gput:Nne
56 \g__tag_struct_dest_num_prop
57 {\@currentHref}
58 {\tag_get:n{struct_num}}
59 }
60 }
61 \AssignSocketPlug{tagsupport/recordtarget}{kernel}
62 \ExplSyntaxOff
```

### 2.1.3 Tagging sockets for toc

`\oc/contentsline/before (socket)` Tagging sockets at the begin and end of contentsline. They receive *all* contentsline arguments as one argument in four brace groups. The socket code should then use the parts it needs.

```
63 \NewSocket{tagsupport/toc/contentsline/before}{1}
64 \NewSocket{tagsupport/toc/contentsline/after}{1}
```

`\ort/toc/starttoc/before (socket)` Tagging sockets for the begin and end of start of `\@starttoc`. They take one argument, `\ort/toc/starttoc/after (socket)` the extension.

```
65 \NewSocket{tagsupport/toc/starttoc/before}{1}
66 \NewSocket{tagsupport/toc/starttoc/after}{1}
```

`\ort/toc/leaders/before (socket)` Tagging sockets to make the dot leaders an artifact. They do not take an argument.

```
67 \NewSocket{tagsupport/toc/leaders/before}{0}
68 \NewSocket{tagsupport/toc/leaders/after}{0}
```

### 2.1.4 Tagging support for table/tabular packages

The code uses a number of sockets to inject the tagging commands. These can be easily set to a noop-plug in case the automated tagging is not wanted.

`\tagsupport/tbl/cell/begin (socket)` At first sockets for the begin and end of cells and table rows:

```
69 \NewSocket{tagsupport/tbl/cell/begin}{0}
70 \NewSocket{tagsupport/tbl/cell/end}{0}
71 \NewSocket{tagsupport/tbl/pcell/begin}{0}
72 \NewSocket{tagsupport/tbl/pcell/end}{0}
```

Multi-line cells have their own sockets (as they start out in vertical mode and need different treatment).

```
73 \NewSocket{tagsupport/tbl/pcell/begin}{0}
74 \NewSocket{tagsupport/tbl/pcell/end}{0}
```

`tagsupport/tbl/init (socket)` This socket should be at the begin of the table, inside a group. It is used for settings such as disabling para-tagging inside the table. This socket can perhaps be merged later into the begin-sockets.

75 \NewSocket{tagsupport/tbl/init}{0}

`port/tbl/init/celldata (socket)` This socket is used in `\tbl_init_cell_data_for_table`, the command that stores and initialize cell data to handle nested tables. It can be used to restore similar tagging related values

76 \NewSocket{tagsupport/tbl/init/celldata}{0}

`agsupport/tbl/finalize (socket)` To fine tune the structure (change cells to header cells, remove unwanted structures, move a foot to the end, etc.). We also need a socket that is executed at the end of the table but *before* all the variables are restored to the outer or default values. The code in the socket can make assignments, but probably shouldn't do typesetting and not write whatsits.

77 \NewSocket{tagsupport/tbl/finalize}{0}

`t/tbl/restore/celldata (socket)` This socket is used in `\tbl_restore_outer_cell_data:`, the command that restores cell data when quitting a nested table. It can be used to restore similar tagging related values

78 \NewSocket{tagsupport/tbl/restore/celldata}{0}

`tagsupport/tbl/colspan (socket)` This socket is used to manage spanning cells, e.g., a `\multicolumn`. It expects one argument (the number of cells spanned) and if tagging is enabled set appropriate tag attributes in the background. We probably need a similar socket for row spans eventually.

79 \NewSocket{tagsupport/tbl/colspan}{1}

`support/tbl/hmode/begin (socket)` These sockets are used in the begin and end code of environments, to allow a fast enabling

`ngsupport/tbl/hmode/end (socket)` and disabling of the tagging. We distinguish between tables that can be used inside

`support/tbl/vmode/begin (socket)` paragraphs and standalone tables such as `longtable` that are always in vertical mode.

`ngsupport/tbl/vmode/end (socket)`

80 \NewSocket{tagsupport/tbl/hmode/begin}{0}

81 \NewSocket{tagsupport/tbl/hmode/end}{0}

82 \NewSocket{tagsupport/tbl/vmode/begin}{0}

83 \NewSocket{tagsupport/tbl/vmode/end}{0}

`port/tbl/longtable/init (socket)` `longtable` needs its own sockets to fine tune the structure. Simply switching the plug

`port/tbl/longtable/finalize (socket)` in the previous socket interferes with enabling/disabling the tagging.

84 \NewSocket{tagsupport/tbl/longtable/init}{0}

85 \NewSocket{tagsupport/tbl/longtable/finalize}{0}

`port/tbl/longtable/head (socket)` Header and footer boxes need special handling because they are repeatedly used.

`port/tbl/longtable/foot (socket)`

86 \NewSocket{tagsupport/tbl/longtable/head}{0}

87 \NewSocket{tagsupport/tbl/longtable/foot}{0}

`port/tbl/leaders/begin (socket)` Sockets around leaders such as rules or dotted lines, that should be tagged as artifacts,

`support/tbl/leaders/end (socket)` used, for example, in `\cline`.

88 \NewSocket{tagsupport/tbl/leaders/begin}{0}

89 \NewSocket{tagsupport/tbl/leaders/end}{0}

### 2.1.5 Tagging Support for floats

```
port/float/hmode/begin (socket) These sockets are used if the float is called in hmode.
support/float/hmode/end (socket) 90 \NewSocket{tagsupport/float/hmode/begin}{0}
91 \NewSocket{tagsupport/float/hmode/end}{0}

tagsupport/float/begin (socket) These sockets start and stop the float structure.
tagsupport/float/end (socket) 92 \NewSocket{tagsupport/float/begin}{0}
93 \NewSocket{tagsupport/float/end}{0}

gsupport/caption/begin (socket) These sockets are used in \@makecaption. They open and close the Caption structure.
tagsupport/caption/end (socket) Their default plugs assume that they are used in vmode. The argument of the begin
socket is the structure number of the parent float. If it is empty the current structure
number is used.
94 \NewSocket{tagsupport/caption/begin}{1}
95 \NewSocket{tagsupport/caption/end}{0}

rt/caption/label/begin (socket) These sockets are used in \@makecaption around the label. Their default plugs ensure
port/caption/label/end (socket) that the label is outside the paragraph and that the rest of the caption uses flattened para
mode. If the caption is not in a hbox, the para/begin socket should follow to properly
start the paragraph.
96 \NewSocket{tagsupport/caption/label/begin}{0}
97 \NewSocket{tagsupport/caption/label/end}{0}
```

## 3 For lttab.dtx parked here for now

```
98 <@@=tbl>
99 \ExplSyntaxOn
```

### 3.1 Variables for row, column and span counting

This part needs a decision on names for various integer registers as well as a decision if those should be also made available for L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> -style packages in form of 2e names and or as non-internals for the L3 programming layer.

At the moment they are all internal but this probably has to change.

```
\g__tbl_col_int
\g__tbl_row_int
\g__tbl_span_tl
\g__tbl_table_cols tl
```

`\g__tbl_row_int` holds the current row number in the table. The value 0 means we haven't yet processed the table preamble (or in case of longtable are just in front of the next chunk to be processed). It is incremented by every `\cr` including the one ending the table preamble.

TODO: due to the gymnastics needed inside the longtable code the row counter is directly exposed there rather than hidden by interfaces. This needs changing when it is decided how to manage these counters.

`\g__tbl_col_int` holds the current column number. The value 0 means we have not yet started the table or just finished a table row (with `\&` typically); any other positive value means we are currently typesetting a cell in that column in some row (denoted by the `\g__tbl_row_int`).

In a `\multicolumn` it holds the column number of the first spanned column and `\g__tbl_span_tl` the info how many cells are spanned.

`\g__tbl_span_tl` is normally 1 except in a `\multicolumn` cell.

```
100 \int_new:N \g__tbl_col_int
```

```

101 \int_new:N \g__tbl_row_int
102 \tl_new:N \g__tbl_span_tl
103 \tl_new:N \g__tbl_table_cols_tl
104
105 \tl_gset:Nn \g__tbl_span_tl {1}
106 \tl_gset:Nn \g__tbl_table_cols_tl {0} % indicates outer level
(End of definition for \g__tbl_col_int and others.)

```

\l\_\_tbl\_saved\_col\_tl Saving the outer values if we are nesting tables is necessary (as the above variables are globally altered). For this we always use token lists because they don't change and we do not need to blow additional integer registers.

```

\l__tbl_saved_row_tl
\l__tbl_saved_span_tl
\l__tbl_saved_table_cols_tl
107 \tl_new:N \l__tbl_saved_col_tl
108 \tl_new:N \l__tbl_saved_row_tl
109 \tl_new:N \l__tbl_saved_span_tl
110 \tl_new:N \l__tbl_saved_table_cols_tl
111
112 \tl_set:Nn \l__tbl_saved_col_tl{0}
113 \tl_set:Nn \l__tbl_saved_row_tl{0}
114 \tl_set:Nn \l__tbl_saved_span_tl{1}
115 \tl_set:Nn \l__tbl_saved_table_cols_tl{0} % indicates outer level
(End of definition for \l__tbl_saved_col_tl and others.)

```

\g\_\_tbl\_missingcells\_int This will contain the number of missing cells in a row:

```

116 \int_new:N \g__tbl_missing_cells_int
(End of definition for \g__tbl_missingcells_int.)

```

### 3.2 Tracing/debugging

```

\DebugTablesOn
\DebugTablesOff
117 \def\DebugTablesOn{
118 \cs_set_eq:NN __tbl_trace:n \typeout
119 }
120 \def\DebugTablesOff{
121 \cs_set_eq:NN __tbl_trace:n \use_none:n
122 }
123 \cs_new_eq:NN __tbl_trace:n \use_none:n
(End of definition for \DebugTablesOn and \DebugTablesOff.)

```

### 3.3 Interface commands

All interface commands for the cell number determination have to be public on some level because they are needed in other packages as well, e.g., longtable. We may or may not also want to provide 2e style names for them.

\tbl\_update\_cell\_data: Updating cell data in columns after the first means we have to increment the \g\_\_tbl\_col\_int by the span count of the previous cell (in case it was a \multicolumn) and then reset the \g\_\_tbl\_span\_tl to one (as the default).

```

124 \cs_new_protected:Npn \tbl_update_cell_data: {
125 \int_gadd:Nn \g__tbl_col_int { \g__tbl_span_tl }
126 \tl_gset:Nn \g__tbl_span_tl {1}
127 }

```

(End of definition for \tbl\_update\_cell\_data:.)

- \tbl\_count\_table\_cols: Current implementation of \@mkpream uses the scratch counter \count@ to keep track of the number of toks registers it needs (2 per column), but this can't be used as it counts also insertions made with !{} and @{}. So similar as does longtable for \LT@cols we count the numbers of ampersands instead.

```
128 \cs_new:Npn \tbl_count_table_cols: {
129 \seq_set_split:NnV\l__tbl_tmpa_seq {&}\@preamble
130 \tl_gset:Ne \g__tbl_table_cols_tl { \seq_count:N \l__tbl_tmpa_seq }
131 __tbl_trace:n { ==>~ Table~ has~ \g__tbl_table_cols_tl \space columns }
132 }
```

(End of definition for \tbl\_count\_table\_cols:.)

\l\_\_tbl\_tmpa\_seq

```
133 \seq_new:N \l__tbl_tmpa_seq
```

(End of definition for \l\_\_tbl\_tmpa\_seq.)

- \tbl\_count\_missing\_cells:n We might have the situation that some table package has not implemented the \tbl\_count\_table\_cols: in which case \g\_\_tbl\_table\_cols\_tl would always be zero and we would get an error below when we try to determine the missing cells, so bypass that calculation if we aren't doing tagging (there the packages should have the proper code added). Recall that this is code, that is called by \\ and an old table packagee might rely on whatever the L<sup>A</sup>T<sub>E</sub>X kernel offers here.

```
134 \cs_new:Npn \tbl_count_missing_cells:n #1 {
135 \tag_if_active:T {
136 \int_compare:nNnT \g__tbl_col_int > 0
137 {
138 \int_gset:Nn \g__tbl_missing_cells_int
139 {
140 \g__tbl_table_cols_tl
141 - \g__tbl_col_int
142 - \g__tbl_span_tl
143 + 1
144 }
145 \int_compare:nNnT \g__tbl_missing_cells_int < 0 \ERROR{missing cells} % should not happen
146 __tbl_trace:n{==>~ (#1)~}
147 This~ row~ needs~
148 \int_use:N \g__tbl_missing_cells_int \space
149 additional~ cell(s)
150 }
151 }
152 }
153 }
154 }
```

(End of definition for \tbl\_count\_missing\_cells:n.)

\tbl\_save\_outer\_table\_cols:

```
155 \cs_new_protected:Npn \tbl_save_outer_table_cols: {
156 \tl_set_eq:NN \l__tbl_saved_table_cols_tl \g__tbl_table_cols_tl
157 }
```

(End of definition for \tbl\_save\_outer\_table\_cols:.)

```
\tbl_init_cell_data_for_table:
158 \cs_new_protected:Npn \tbl_init_cell_data_for_table: {
159 \tl_set:NNo \l__tbl_saved_col_tl {\int_use:N \g__tbl_col_int }
160 \tl_set:NNo \l__tbl_saved_row_tl {\int_use:N \g__tbl_row_int }
161 \tl_set_eq:NN \l__tbl_saved_span_tl \g__tbl_span_tl
162 %
163 __tbl_trace:n { ==>~ saved-cell-data:~
164 \l__tbl_saved_row_tl,
165 \l__tbl_saved_col_tl,
166 \l__tbl_saved_span_tl \space
167 (
168 \int_compare:nNnTF \l__tbl_saved_table_cols_tl = 0
169 { outer~ level }
170 { max:~ \l__tbl_saved_table_cols_tl }
171)
172 }
173 }
```

Tagging has to initialize cell data too.

```
173 \UseTaggingSocket{tbl/init/celldata}
```

These are the initial values when starting a table:

```
174 \int_gzero:N \g__tbl_row_int
175 \int_gzero:N \g__tbl_col_int
176 \tl_gset:Nn \g__tbl_span_tl {1}
177 }
```

(End of definition for \tbl\_init\_cell\_data\_for\_table:.)

\tbl\_update\_cell\_data\_for\_next\_row:

```
178 \cs_new_protected:Npn \tbl_update_cell_data_for_next_row: {
179 \int_gincr:N \g__tbl_row_int % this row about to start
180 \int_gzero:N \g__tbl_col_int % we are before first col
181 }
```

(End of definition for \tbl\_update\_cell\_data\_for\_next\_row:.)

\tbl\_init\_cell\_data\_for\_row: If we start processing a cell in the first column we set \g\_\_tbl\_col\_int to 1 as we are no longer "at" but "in" the first column. We also set \g\_\_tbl\_span\_tl to its default value (not spanning cells).

```
182 \cs_new_protected:Npn \tbl_init_cell_data_for_row: {
183 \int_gset:Nn \g__tbl_col_int {1}
184 \tl_gset:Nn \g__tbl_span_tl {1}
185 }
```

(End of definition for \tbl\_init\_cell\_data\_for\_row:.)

\tbl\_if\_row\_was\_started:T \tbl\_if\_row\_was\_started:TF We use \g\_\_tbl\_col\_int equal zero to indicate that we are just after a TR (i.e.n between rows or at the very beginning of the table). Using the row count is not so good as longtable may split the table in chunks.

These conditionals have to be expandable (i.e., unprotected) as they are sometimes executed when T<sub>E</sub>X is scanning inside a table.

```
186 \cs_new:Npn \tbl_if_row_was_started:T {
187 \int_compare:nNnT \g__tbl_col_int > 0
```

```

188 }
189 \cs_new:Npn \tbl_if_row_was_started:TF {
190 \int_compare:nNnTF \g__tbl_col_int > 0
191 }

```

(End of definition for \tbl\_if\_row\_was\_started:T and \tbl\_if\_row\_was\_started:TF.)

- \tbl\_gzero\_row\_count: This here is basically a temporary interface. What it will be in the end depends on what we decide concerning exposing row and column counters, if they stay internal we need something like this here (perhaps using `gincr` etc, or perhaps some other names in the first place).

```

192 \cs_new_protected:Npn \tbl_gzero_row_count: {
193 \int_gzero:N \g__tbl_row_int
194 }
195 \cs_new_protected:Npn \tbl_gincr_row_count: {
196 \int_gincr:N \g__tbl_row_int
197 }
198 \cs_new_protected:Npn \tbl_gdecr_row_count: {
199 \int_gdecr:N \g__tbl_row_int
200 }

```

(End of definition for \tbl\_gzero\_row\_count:, \tbl\_gincr\_row\_count:, and \tbl\_gdecr\_row\_count::)

- \tbl\_inbetween\_rows: Again name is not really brilliant so far.

```

201 \cs_new_protected:Npn \tbl_inbetween_rows: {
202 \int_gzero:N \g__tbl_col_int
203 }

```

(End of definition for \tbl\_inbetween\_rows::)

- \tbl\_restore\_outer\_cell\_data:

```

204 \cs_new_protected:Npn \tbl_restore_outer_cell_data: {
205 \int_gset:Nn \g__tbl_col_int { \l__tbl_saved_col_tl }
206 \int_gset:Nn \g__tbl_row_int { \l__tbl_saved_row_tl }
207 \tl_gset_eq:NN \g__tbl_span_tl \l__tbl_saved_span_tl
208 \tl_gset_eq:NN \g__tbl_table_cols_tl \l__tbl_saved_table_cols_tl
209 \UseTaggingSocket{tbl/restore/celldata}
210 __tbl_trace:n { ==>~ restored~cell~data:~ }
211 \int_use:N \g__tbl_row_int,
212 \int_use:N \g__tbl_col_int,
213 \l__tbl_saved_span_tl \space
214 (
215 \int_compare:nNnTF \g__tbl_table_cols_tl = 0
216 { outer~ level }
217 { max:~ \g__tbl_table_cols_tl }
218)
219 }
220 }

```

(End of definition for \tbl\_restore\_outer\_cell\_data::)

- \tbl\_update\_multicolumn\_cell\_data:n This macro updates `\g__tbl_col_int` and `\g__tbl_span_tl` inside a `\multicolumn` and possibly calls the tagging socket `tbl/row/begin`.

```

221 \cs_new_protected:Npn \tbl_update_multicolumn_cell_data:n #1 {

```

We execute socket for tagging only if this `\multicolumn` replaces the preamble of the first column. In that case we also have to set `\g__tbl_col_int` to 1 because this is no longer done in the preamble for the cell either.

```

222 \int_compare:nNnTF \g__tbl_col_int = 0
223 {
224 \UseTaggingSocket{tbl/row/begin}
225 \int_gset:Nn \g__tbl_col_int {1}
226 }

```

If we are in a later column we use `\g__tbl_span_tl` from the previous column to update.

```

227 {
228 \int_gadd:Nn \g__tbl_col_int { \g__tbl_span_tl }
229 }

```

Then we set the span value so that it can be used in the next column.

```

230 \tl_gset:Nn \g__tbl_span_tl {#1}
231 }

```

*(End of definition for `\tbl_update_multicolumn_cell_data:n`.)*

**\tbl\_cr\_cr:n** This macro is used instead of the usual `\cr\cr` at the end of a table. It is deliberately defined without protection because it may get expanded by the scanning mechanism of low-level TeX after a final `\cr` (aka `\।`) in the table. In that case it shouldn't stop the expansion and the conditional inside will be false, thus it just vanishes without doing anything. If there are missing cells (in which case we also haven't seen `\cr` yet) the macro `\tbl_count_missing_cells:n` is executed and then the row is finished with a final `\cr`.

```

232 \cs_new:Npn \tbl_cr_cr:n #1 {
233 \int_compare:nNnT \g__tbl_col_int > 0
234 {
235 \tbl_count_missing_cells:n {#1}
236 }

```

Even if we are at the start of a row we may have to do a `\cr`, so we do a `\cr\cr` always at the end.

```

237 \cr\cr
238 }

```

*(End of definition for `\tbl_cr_cr:n`.)*

```

239 \ExplSyntaxOff
240
```

This is needed for `longtable` because `\refstepcounter` is setting up a target when `hyperref` is loaded and we don't want that in `longtable`. Prevent `longtable` patching by `hyperref` until `hyperref` does so automatically:

```

241 \def\hyper@nopatch@longtable{}
Should there be a module?
242 ⟨latexrelease⟩\NewModuleRelease{2024/06/01}{ltagging}
243 ⟨latexrelease⟩ {Tagging support}
244 ⟨latexrelease⟩\IncludeInRelease{0000/00/00}{ltagging}%
245 ⟨latexrelease⟩ {Undo tagging support}
246 ⟨latexrelease⟩
247 ⟨latexrelease⟩
248 ⟨latexrelease⟩
249 ⟨latexrelease⟩\EndModuleRelease
250 ⟨/2ekernel | latexrelease⟩

```

## File 56

### lthyphen.dtx

This file contains the code for loading hyphenation patterns into L<sup>A</sup>T<sub>E</sub>X. Most of this will end up in a file called `hyphen.ltx`. If you wish to customize your L<sup>A</sup>T<sub>E</sub>X system in respect of hyphenation patterns, write a file `hyphen.cfg`. If this file exists, it will be loaded instead of `hyphen.ltx`. See the comments below for additional information.

To produce the printed version of this file the following code is used. It can be extracted with the `DOCSTRIP` program, or one can run this file directly through L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub> .

```
1 <*driver>
2 \documentclass{ltxdoc}
3 \begin{document}
4 \DocInput{lthyphen.dtx}
5 \end{document}
6 </driver>
```

The default file `hyphen.ltx` loads hyphenation patterns for US english. If you want to load additional or other hyphenation patterns, you should create a file `hyphen.cfg`. This is best done by starting from `hyphen.ltx`.

For backward compatibility, the default file, `hyphen.ltx`, first tries to load the file `hyphen.tex`. If this file exists, an information message is issued and the appropriate defaults for T<sub>E</sub>X's internal parameters are set: `\language` is initialized to 0, and `\lefthyphenmin` and `\righthyphenmin` to 2 and 3, respectively, to disallow x- or -xx breaks.

```
7 <*default>
8 \InputIfFileExists{hyphen.tex}%
9 {\message{Loading hyphenation patterns for US english.}%
10 \language=0
11 \lefthyphenmin=2 \righthyphenmin=3 }%
```

Otherwise, since we cannot do anything without any hyphenation patterns, an error message is printed and the IniT<sub>E</sub>X run is terminated by invoking `\@@end` (which is the L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>  name for T<sub>E</sub>X's `\end` primitive).

```
12 {\errhelp{The configuration for hyphenation is incorrectly
13 installed.^^J%
14 If you don't understand this error message you need
15 to seek^Jexpert advice.}%
16 \errmessage{OOPS! I can't find any hyphenation patterns for
17 US english.^^J \space Think of getting some or the
18 latex2e setup will never succeed}\@@end}
19 </default>
```

The following example describes the possible contents of a file `hyphen.cfg` that will load both US English and German hyphenation patterns, making the former the default. It sets `\language` to 0 for the US patterns and to 1 for the German patterns. Then `\language` is set to 0 to make this the default and the default values of `\lefthyphenmin` and `\righthyphenmin` are set.

```
language=0
input hyphen % (or \input ushyphen1 if the file has been renamed)
language=1
input ghyph31
```

```
language=0
lefthyphenmin=2
righthyphenmin=3
endinput
```

Another possibility is to use the package `babel`, by Johannes Braams. That package is distributed with a suitable `hyphen.cfg` file.

# File 57

## ltfinal.dtx

### 1 Final settings

This section contains the final settings for L<sup>A</sup>T<sub>E</sub>X. It initializes some debugging and typesetting parameters, sets the default \catcodes and uc/lc codes, and inputs the hyphenation file.

#### 1.1 Debugging

By default, L<sup>A</sup>T<sub>E</sub>X shows statistics:

```
1 <*2ekernel>
2 \tracingstats1
```

#### 1.2 Typesetting parameters

\@lowpenalty These are penalties used internally.  
\@medpenalty  
\@highpenalty

```
3 \newcount\@lowpenalty
4 \newcount\@medpenalty
5 \newcount\@highpenalty
```

(End of definition for \@lowpenalty, \@medpenalty, and \@highpenalty.)

\newmarks Allocate extended marks types if etex is active. Placed here at the end of the format to increase compatibility with count allocations in earlier releases.

```
6 </2ekernel>
7 <*2ekernel | latexrelease>
8 <latexrelease>\IncludeInRelease{2015/01/01}%
9 <latexrelease> {\newmarks}{Extended Allocation}%
10 \ifx\marks\@undefined\else
11 \def\newmarks{%
12 \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\@alloc@top}
13 \fi
14 </2ekernel | latexrelease>
15 <latexrelease>\EndIncludeInRelease
16 <latexrelease>\IncludeInRelease{0000/00/00}%
17 <latexrelease> {\newmarks}{Extended Allocation}%
18 <latexrelease>\let\newmarks\@undefined
19 <latexrelease>\EndIncludeInRelease
20 <*2ekernel>
```

(End of definition for \newmarks.)

Allocate 3 mark classes to be used in \markboth and \markright. Should be done earlier but for that definition of \newmarks needs moving (which it should I guess).

```
21 </2ekernel>
22 <*2ekernel | latexrelease>
23 <latexrelease>\IncludeInRelease{2022/06/01}%
24 <latexrelease> {2e-left}{Delayed legacy marks}%
25 \NewMarkClass {2e-left}
```

```

26 \NewMarkClass {2e-right}
27 \NewMarkClass {2e-right-nonempty}
No rollback really, the marks will remain.
28 </2ekernel | latexrelease>
29 <latexrelease>\EndIncludeInRelease
30 <latexrelease>\IncludeInRelease{0000/00/00}%
31 <latexrelease> {2e-left}{Delayed legacy marks}%
32 <latexrelease>
33 <latexrelease>\EndIncludeInRelease
34 <*2ekernel>

\newXeTeXintercharclass Allocate \XeTeXintercharclass types if xetex is active. previously defined in xetex.ini.
\xe@alloc@intercharclass
\@alloc@intercharclass@top
35 </2ekernel>
36 <*2ekernel | latexrelease>
37 <latexrelease>\IncludeInRelease{2015/01/01}%
38 <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).
39 \ifx\XeTeXcharclass\@undefined
40 \else
41 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
42 \chardef\xe@alloc@intercharclass@top=4095
43 \else
44 \chardef\xe@alloc@intercharclass@top=255
45 \fi
46 \def\newXeTeXintercharclass{%
47 \xe@alloc\XeTeXcharclass
48 \chardef\xe@alloc@intercharclass\m@ne\xe@alloc@intercharclass@top}
49 \fi
50 </2ekernel | latexrelease>
51 <latexrelease>\EndIncludeInRelease
52 <latexrelease>\IncludeInRelease{0000/00/00}%
53 <latexrelease> {\newXeTeXintercharclass}{Extended Allocation}%
54 <latexrelease> \ifx\XeTeXcharclass\@undefined
55 <latexrelease> \else
56 <latexrelease> \def\xe@alloc@#1#2#3#4{\global\advance#1\@ne
57 <latexrelease> \xe@ch@ck#1#4#2%
58 <latexrelease> \allocationnumber#1%
59 <latexrelease> \global#3#5\allocationnumber
60 <latexrelease> \wlog{\string#5=\string#2\the\allocationnumber}}
61 <latexrelease> \def\xe@ch@ck#1#2#3{%
62 <latexrelease> \ifnum#1<#2\else
63 <latexrelease> \errmessage{No room for a new #3}%
64 <latexrelease> \fi}
65 <latexrelease> \def\newXeTeXintercharclass{%
66 <latexrelease> \xe@alloc@\xe@alloc@intercharclass
67 <latexrelease> \XeTeXcharclass\chardef\@ccilv}
68 <latexrelease> \fi
69 <latexrelease>\EndIncludeInRelease
70 <*2ekernel | latexrelease>
71 <latexrelease>\IncludeInRelease{2016/02/01}%

```

```

72 〈\latexrelease〉 {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
73 \ifx\XeTeXcharclass\@undefined
74 \else
75 \countdef\xe@alloc@intercharclass=257
76 \xe@alloc@intercharclass=\z@
77 \fi
78 〈/2ekernel | \latexrelease〉
79 〈\latexrelease〉\EndIncludeInRelease
80 〈\latexrelease〉\IncludeInRelease{2015/01/01}%
81 〈\latexrelease〉 {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
82 〈\latexrelease〉 \ifx\XeTeXcharclass\@undefined
83 〈\latexrelease〉 \else
84 〈\latexrelease〉 \xe@alloc@intercharclass=\thr@@
85 〈\latexrelease〉 \fi
86 〈\latexrelease〉\EndIncludeInRelease
87 〈\latexrelease〉\IncludeInRelease{0000/00/00}%
88 〈\latexrelease〉 {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
89 〈\latexrelease〉 \ifx\XeTeXcharclass\@undefined
90 〈\latexrelease〉 \else
91 〈\latexrelease〉 \newcount\xe@alloc@intercharclass
92 〈\latexrelease〉 \xe@alloc@intercharclass=\thr@@
93 〈\latexrelease〉 \fi
94 〈\latexrelease〉\EndIncludeInRelease
95 〈*2ekernel〉

(End of definition for \newXeTeXintercharclass, \xe@alloc@intercharclass, and
 \e@alloc@intercharclass@top.)

```

`trace_stack_levels` Now define the Lua function to emulate `\tracingstacklevels` and install it in the `input_level_string` callback.

```

96 〈/2ekernel〉
97 〈*2ekernel | \latexrelease〉

```

In `\latexrelease` mode we always remove the function from the callback, then add the correct version later.

```

98 〈\latexrelease〉\ifx\directlua\@undefined
99 〈\latexrelease〉\else
100 〈\latexrelease〉 \directlua{%
101 \if luatexbase.callbacktypes['input_level_string'] and %
102 luatexbase.in_callback('input_level_string','tracingstacklevels') then
103 luatexbase.remove_from_callback('input_level_string','tracingstacklevels')
104 end}%
105 〈\latexrelease〉\fi
106 〈\latexrelease〉\IncludeInRelease{2021/06/01}{trace_stack_levels}%
107 〈\latexrelease〉 {Lua trace_stack_levels function}%
108 〈\ifx\directlua\@undefined
109 〈\else
110 〈*2ekernel〉
111 〈\expanded{%
112 \everyjob{\the\everyjob
113 \noexpand}\directlua
114 〈/2ekernel〉
115 \directlua{%
116 local function trace_stack_levels (input_ptr)
117 local tracingstacklevels = tex.count.tracingstacklevels

```

```

118 if tex.tracingmacros > 0 or input_ptr < tracingstacklevels then
119 if tracingstacklevels > 0 then
120 if input_ptr < tracingstacklevels then
121 return "\string\n\string~" .. string.rep(".",input_ptr)
122 else
123 return "\string~\string~"
124 end
125 else
126 return "\string\n"
127 end
128 else
129 return ""
130 end
131 end
132 \laterelease if luatexbase.callbacktypes['input_level_string'] then
133 luatexbase.add_to_callback('input_level_string',
134 trace_stack_levels,'tracingstacklevels')
135 \laterelease end
136 }%
137 {*2ekernel}
138 }}%
139 /2ekernel}
140 \fi
141 \laterelease\EndIncludeInRelease
142 \laterelease

```

Then for the full rollback, just do nothing, since the function was already taken out of the rollback above.

```

143 \laterelease\IncludeInRelease{0000/00/00}{trace_stack_levels}%
144 \laterelease {Lua trace_stack_levels function}%
145 \laterelease% Nothing here
146 \laterelease\EndIncludeInRelease
147 /2ekernel | latexrelease}
148 {*2ekernel}

```

*(End of definition for trace\_stack\_levels.)*

The default values of the picture and \fbox parameters:

```

149 \unitlength = 1pt
150 \fboxsep = 3pt
151 \fboxrule = .4pt

```

The saved value of TeX's \maxdepth:

```

152 \cmaxdepth = \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}

```

153 \vsize = 1000pt
154 \@colroom = \vsize
155 \@colht = \vsize

```

Initialise \textheight \textwidth and page style, to avoid internal errors if they are not set by the class.

```

156 \textheight=.5\maxdimen
157 \textwidth=\textheight
158 \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.

```

159 \ifnum 0%
160 \ifx\Umathcode\@undefined\else 1\fi
161 \ifx\XeTeXmathcode\@undefined\else 1\fi
162 >\z@
163 \message{ Unicode character data,}
164 \input{load-unicode-data}
165 {/2ekernel}
166 \langle latexrelease \rangle \IncludeInRelease{2016/02/01}%
167 \langle latexrelease \rangle {\XeTeXintercharclasses}{XeTeX character classes}%
168 \langle latexrelease \rangle \ifx\XeTeXinterchartoks\undefined
169 \langle latexrelease \rangle \else
170 \langle latexrelease \rangle \begingroup
171 \langle latexrelease \rangle \chardef\XeTeXcharclassID = 0 %
172 \langle latexrelease \rangle \chardef\XeTeXcharclassOP = 0 %
173 \langle latexrelease \rangle \chardef\XeTeXcharclassCL = 0 %
174 \langle latexrelease \rangle \chardef\XeTeXcharclassEX = 0 %
175 \langle latexrelease \rangle \chardef\XeTeXcharclassIS = 0 %
176 \langle latexrelease \rangle \chardef\XeTeXcharclassNS = 0 %
177 \langle latexrelease \rangle \chardef\XeTeXcharclassCM = 0 %
178 \langle latexrelease \rangle \input{load-unicode-xetex-classes}
179 \langle latexrelease \rangle \endgroup
180 \langle latexrelease \rangle \global\let\xtxHanGlue\undefined
181 \langle latexrelease \rangle \global\let\xtxHanSpace\undefined
182 \langle latexrelease \rangle \global\XeTeXinterchartoks 0 1 = {}
183 \langle latexrelease \rangle \global\XeTeXinterchartoks 0 2 = {}
184 \langle latexrelease \rangle \global\XeTeXinterchartoks 0 3 = {}
185 \langle latexrelease \rangle \global\XeTeXinterchartoks 1 0 = {}
186 \langle latexrelease \rangle \global\XeTeXinterchartoks 2 0 = {}
187 \langle latexrelease \rangle \global\XeTeXinterchartoks 3 0 = {}
188 \langle latexrelease \rangle \global\XeTeXinterchartoks 1 1 = {}
189 \langle latexrelease \rangle \global\XeTeXinterchartoks 1 2 = {}
190 \langle latexrelease \rangle \global\XeTeXinterchartoks 1 3 = {}
191 \langle latexrelease \rangle \global\XeTeXinterchartoks 2 1 = {}
192 \langle latexrelease \rangle \global\XeTeXinterchartoks 2 2 = {}
193 \langle latexrelease \rangle \global\XeTeXinterchartoks 2 3 = {}
194 \langle latexrelease \rangle \global\XeTeXinterchartoks 3 1 = {}
195 \langle latexrelease \rangle \global\XeTeXinterchartoks 3 2 = {}
196 \langle latexrelease \rangle \global\XeTeXinterchartoks 3 3 = {}
197 \langle latexrelease \rangle \fi
198 \langle latexrelease \rangle \EndIncludeInRelease
199 \langle latexrelease \rangle \IncludeInRelease{0000/00/00}%

```

```

200 <{latexrelease}> {\XeTeXintercharclasses}{XeTeX character classes}%
201 <{latexrelease}> \ifx\XeTeXinterchartoks\undefined
202 <{latexrelease}> \else
203 <{latexrelease}> \input{load-unicode-xetex-classes}
204 <{latexrelease}> \gdef\xtxHanGlue{\hskip0pt plus 0.1em\relax}
205 <{latexrelease}> \gdef\xtxHanSpace{\hskip0.2em plus 0.2em minus 0.1em\relax}
206 <{latexrelease}> \global\XeTeXinterchartoks 0 1 = {\xtxHanSpace}
207 <{latexrelease}> \global\XeTeXinterchartoks 0 2 = {\xtxHanSpace}
208 <{latexrelease}> \global\XeTeXinterchartoks 0 3 = {\nobreak\xtxHanSpace}
209 <{latexrelease}> \global\XeTeXinterchartoks 1 0 = {\xtxHanSpace}
210 <{latexrelease}> \global\XeTeXinterchartoks 2 0 = {\nobreak\xtxHanSpace}
211 <{latexrelease}> \global\XeTeXinterchartoks 3 0 = {\xtxHanSpace}
212 <{latexrelease}> \global\XeTeXinterchartoks 1 1 = {\xtxHanGlue}
213 <{latexrelease}> \global\XeTeXinterchartoks 1 2 = {\xtxHanGlue}
214 <{latexrelease}> \global\XeTeXinterchartoks 1 3 = {\nobreak\xtxHanGlue}
215 <{latexrelease}> \global\XeTeXinterchartoks 2 1 = {\nobreak\xtxHanGlue}
216 <{latexrelease}> \global\XeTeXinterchartoks 2 2 = {\nobreak\xtxHanGlue}
217 <{latexrelease}> \global\XeTeXinterchartoks 2 3 = {\xtxHanGlue}
218 <{latexrelease}> \global\XeTeXinterchartoks 3 1 = {\xtxHanGlue}
219 <{latexrelease}> \global\XeTeXinterchartoks 3 2 = {\xtxHanGlue}
220 <{latexrelease}> \global\XeTeXinterchartoks 3 3 = {\nobreak\xtxHanGlue}
221 <{latexrelease}> \fi
222 <{latexrelease}>\EndIncludeInRelease
223 <{*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.

```
224 \lccode`-='`- % default hyphen char
```

The alternative is that a “traditional” engine is in use.

```
225 \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.

```

226 \def\reserved@a#1#2{%
227 @_tempcnta#1\relax
228 @_tempcntb#2\relax
229 \reserved@b
230 }
231 \def\reserved@b{%
232 \ifnum @_tempcnta> @_tempcntb\else
233 \reserved@c @_tempcnta
234 \advance @_tempcnta \ne
235 \expandafter \reserved@b
236 \fi
237 }
```

Depending on the T<sub>E</sub>X version, we might not be allowed to do this for non-ASCII characters.

```

238 \def\reserved@c#1{%
239 \count@=#1\advance \count@ by -"20
240 \uccode#1=\count@
241 \lccode#1=#1
242 }
```

```

243 \reserved@{\`a}{\`z}
244 \reserved@{"A0}{ "BC}
245 \reserved@{"E0}{ "FF}

```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcodes` set to 999.

```

246 \def\reserved@c#1{%
247 \count@=\#1\advance\count@ by "20
248 \uccode#1=\#1
249 \lccode#1=\count@
250 \sfcodes#1=999
251 }
252 \reserved@{\`A}{\`Z}
253 \reserved@{"80}{ "9C}
254 \reserved@{"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.

```

255 \uccode`^Y=\`I % dotless i
256 \lccode`^Y=\^Y % dotless i
257 \uccode`^Z=\`J % dotless j, ae in OT1
258 \lccode`^Z=\^Z % dotless j, ae in OT1
259 \lccode`^9d=\`i % dotted I
260 \uccode`^9d=\^9d % dotted I
261 \lccode`^9e=\^9e % d-bar
262 \uccode`^9e=\^d0 % d-bar

```

Finally here is one that helps hyphenation in the OT1 encoding.

```
263 \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).

```

264 \lccode`\- =`\- % default hyphen char
265 \lccode 127=127 % alternate hyphen char
266 \lccode 23 =23 % textcompwordmark in T1

```

End of the conditional to select either Unicode or T1 encoding defaults.

```
267 \fi
```

At this stage, we can install any last-minute `expl3` set-up.

```

268 \Oexpl@finalise@setup@@
269 \def\@expl@finalise@setup@@{}}

```

This is as good a place as any to active a few XeTeX-specific settings

```

270 \ifx\XeTeXuseglyphmetrics\undefined
271 \else
272 \XeTeXuseglyphmetrics=1 %
273 \XeTeXdashbreakstate=1 %
274 \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 `\catcodes` are set, so local hyphenation files can use 8-bit input.

We try to load the customized hyphenation description file.

```

275 \InputIfFileExists{hyphen.cfg}
276 {\typeout{=====
277 Local configuration file hyphen.cfg used^^J%
278 =====}%
279 \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
280 }
281 {\input{hyphen.ltx}}
282 \let\@addtofilelist\@gobble

\l@nohyphenation
283 \ifx\l@nohyphenation \undefined
284 \newlanguage\l@nohyphenation
285 \fi

(End of definition for \l@nohyphenation.)

```

\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.

```

286 </2ekernel>
287 <*2ekernel | latexrelease>
288 <latexrelease>\IncludeInRelease{2017/04/15}%
289 <latexrelease> {\document@default@language}{Save language for hyphenation}%
290 \let\document@default@language\m@ne
291 </2ekernel | latexrelease>
292 <latexrelease>\EndIncludeInRelease
293 <latexrelease>\IncludeInRelease{0000/00/00}%
294 <latexrelease> {\document@default@language}{Save language for hyphenation}%
295 %
296 <latexrelease>\let\document@default@language\undefined
297 <latexrelease>\EndIncludeInRelease
298 <*2ekernel>

(End of definition for \document@default@language.)

```

## 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.

```

299 \ifdim \font@submax >\z@
300 @font@warning{Size substitutions with differences\MessageBreak
301 up to \font@submax\space have occurred.\MessageBreak
302 \MessageBreak
303 Please check the transcript file
304 carefully\MessageBreak
305 and redo the format generation if necessary!
306 \@gobbletwo}%
307 \errhelp{Only stopped, to give you time to
308 read the above message.}%
309 \errmessage{%

```

We reset the macro. Otherwise every user will get a warning on every job.

```

310 \def\font@submax{0pt}
311 \fi

```

For pdfTeX preload and enable automatic glyph to Unicode mapping for more reliable copy and paste support.

```

312 </2ekernel>
313 <*2ekernel | latexrelease>
314 <latexrelease>\IncludeInRelease{2021/06/01}%
315 <latexrelease> {\pdfgentounicode}{Preload glyptounicode}%
316 \ifx \pdfgentounicode \undefined \else
317 <*2ekernel>
318 \ifnum 0=0%
319 \ifdefined\pdftexversion
320 % \pdftexversion<140 does not have \pdfgentounicode, so we only check higher values
321 \ifnum \pdftexversion=140 \ifnum\pdftexrevision<22 1\fi\fi
322 \fi
323 \relax
324 </2ekernel>
325 \input glyptounicode
326 <*2ekernel>
327 \else
328 \begingroup
329 \everyeof{\noexpand}\endlinechar-1
330 \edef\x{\endgroup
331 \everyjob{\the\everyjob\@input glyptounicode }%
332 }\x
333 \fi
334 </2ekernel>
335 \pdfgentounicode=1
336 \fi
337 </2ekernel | latexrelease>
338 <latexrelease>\EndIncludeInRelease

```

When rolling back we can't unload the glyptounicode mappings, but we can reset `\pdfgentounicode` to ensure that they aren't used.

```

339 <latexrelease>\IncludeInRelease{0000/00/00}%
340 <latexrelease> {\pdfgentounicode}{Preload glyptounicode}%
341 <latexrelease>\ifx \pdfgentounicode \undefined \else
342 <latexrelease> \pdfgentounicode=0
343 <latexrelease>\fi
344 <latexrelease>\EndIncludeInRelease
345 <*2ekernel>

```

## 1.6 Input encoding

Starting with the 2018 L<sup>A</sup>T<sub>E</sub>X 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.

```

346 </2ekernel>
347 <*2ekernel | latexrelease>

```

Check that a classic 8-bit tex engine is being used (LaTeX or PDFLaTeX).

```
348 <latexrelease>\IncludeInRelease{2018/04/01}%
349 <latexrelease> {\UTFviii@invalid}{UTF-8 default}%
```

Skip this section in Unicode TeX, or if MLTeX and EncTeX are enabled.

```
350 \ifnum0%
351 \ifx\Umathcode\@undefined\else 1\fi
352 \ifx\mubyte\@undefined\else 1\fi
353 \ifx\charsubdef\@undefined\else 1\fi
354 =\z@
355 \def\saved@space@catcode{10}
356 \let\@inpenc@test\relax
357 \def\IeC{%
358 \ifx\protect\@typeset@protect
359 \expandafter\@firstofone
360 \else
361 \noexpand\IeC
362 \fi
363 }
```

Make characters active for UTF-8 input formats

```
364 \@tempcnta=1
365 \loop
366 \catcode\@tempcnta=13 %
367 \advance\@tempcnta\@ne %
368 \ifnum\@tempcnta<32 %
369 \repeat %
370 \catcode0=15 % null
371 \catcode9=10 % tab
372 \catcode10=12 % ctrl J
373 \catcode12=13 % ctrl L
374 \catcode13=5 % newline
375 \@tempcnta=128
376 \loop
377 \catcode\@tempcnta=13
378 \advance\@tempcnta\@ne
379 \ifnum\@tempcnta<256
380 \repeat
```

\UseRawInputEncoding Reset 8 bit characters to catcode 12 so the input encoding matches the “Raw” font encoding. Useful for special behaviours, or for compatibility with older L<sup>A</sup>T<sub>E</sub>X formats.

```
381 \def\UseRawInputEncoding{%
382 \let\inputencodingname\@undefined % revert
383 \let\DeclareFontEncoding@\DeclareFontEncoding@saved % revert
384 \let\DeclareUnicodeCharacter\@undefined % revert
385 \@tempcnta=1
386 \loop
387 \catcode\@tempcnta=15 %
388 \advance\@tempcnta\@ne %
389 \ifnum\@tempcnta<32 %
390 \repeat %
391 \catcode0=15 % null
392 \catcode9=10 % tab
```

```

393 \catcode10=12 % ctrl J
394 \catcode12=13 % ctrl L
395 \catcode13=5 % newline
396 \tempcnta=128
397 \loop
398 \catcode\tempcnta=12
399 \advance\tempcnta\@ne
400 \ifnum\tempcnta<256
401 \repeat
402 }

(End of definition for \UseRawInputEncoding.)

```

\DeclareFontEncoding@saved Saved version of \DeclareFontEncoding@ before utf8.def modifies it for use in \UseRawInputEncoding above.

```
403 \let\DeclareFontEncoding@saved\DeclareFontEncoding@
```

(*End of definition for \DeclareFontEncoding@saved.*)

```

404 \edef\inputencodingname{utf8}%
405 \input{utf8.def}
406 \let\UTFviii@undefined@err@@\UTFviii@undefined@err
407 \let\UTFviii@invalid@err@@\UTFviii@invalid@err
408 \let\UTFviii@two@octets@@\UTFviii@two@octets
409 \let\UTFviii@three@octets@@\UTFviii@three@octets
410 \let\UTFviii@four@octets@@\UTFviii@four@octets
411 {2ekernel}\def\UTFviii@undefined@err#1{\@gobble#1}%
412 {2ekernel}\let\UTFviii@invalid@err\string
413 {2ekernel}\let\UTFviii@two@octets\string
414 {2ekernel}\let\UTFviii@three@octets\string
415 {2ekernel}\let\UTFviii@four@octets\string
416 {2ekernel}\everyjob\expandafter{\the\everyjob
417 {2ekernel}\let\UTFviii@undefined@err\UTFviii@undefined@err@@
418 {2ekernel}\let\UTFviii@invalid@err\UTFviii@invalid@err@@
419 {2ekernel}\let\UTFviii@two@octets\UTFviii@two@octets@@
420 {2ekernel}\let\UTFviii@three@octets\UTFviii@three@octets@@
421 {2ekernel}\let\UTFviii@four@octets\UTFviii@four@octets@@
422 {2ekernel}}
423 \let@\inpcnt@test\@undefined
424 \let\@space@catcode\@undefined

```

For formats not set up for UTF-8 default, set the C0 controls to catcode 15.

```

425 \else
426 \tempcnta=0
427 \loop
428 \catcode\tempcnta=15 %
429 \advance\tempcnta\@ne %
430 \ifnum\tempcnta<32 %
431 \repeat %
432 \catcode0=15 % null
433 \catcode9=10 % tab
434 \catcode10=12 % ctrl J
435 \catcode12=13 % ctrl L
436 \catcode13=5 % newline
437 \let\UseRawInputEncoding\relax

```

This ends the skipped code in Unicode engines:

```
438 \fi
439 </2ekernel | latexrelease>
440 <latexrelease>\EndIncludeInRelease
441 <latexrelease>\IncludeInRelease{0000/00/00}%
442 <latexrelease> {\UTFviii@invalid}{UTF-8 default}%
443 <latexrelease> \let\UTFviii@two@octets@combine@\undefined
444 <latexrelease> \let\UTFviii@three@octets@combine@\undefined
445 <latexrelease> \let\UTFviii@four@octets@combine@\undefined
446 <latexrelease> \let\UTFviii@two@octets@string@\undefined
447 <latexrelease> \let\UTFviii@three@octets@string@\undefined
448 <latexrelease> \let\UTFviii@four@octets@string@\undefined
449 <latexrelease> \let\UTFviii@two@octets@noexpand@\undefined
450 <latexrelease> \let\UTFviii@three@octets@noexpand@\undefined
451 <latexrelease> \let\UTFviii@four@octets@noexpand@\undefined
452 <latexrelease> \@tempcnta=0
453 <latexrelease> \loop
454 <latexrelease> \catcode@\tempcnta=15
455 <latexrelease> \advance@\tempcnta\@ne
456 <latexrelease> \ifnum@\tempcnta<32
457 <latexrelease> \repeat %
458 <latexrelease> \catcode9=10 % tab
459 <latexrelease> \catcode10=12 % ctrl J
460 <latexrelease> \catcode12=13 % ctrl L
461 <latexrelease> \catcode13=5 % newline
462 <latexrelease> \@tempcnta=128
463 <latexrelease> \loop
464 <latexrelease> \catcode@\tempcnta=12
465 <latexrelease> \advance@\tempcnta\@ne
466 <latexrelease> \ifnum@\tempcnta<256
467 <latexrelease> \repeat
468 <latexrelease> \let\IeC@\undefined
469 <latexrelease> \def\DeclareFontEncoding@#1#2#3{%
470 <latexrelease> \expandafter
471 <latexrelease> \ifx\csname T@#1\endcsname\relax
472 <latexrelease> \def\cdp@elt{\noexpand\cdp@elt}%
473 <latexrelease> \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
474 <latexrelease> {\default@family}{\default@series}%
475 <latexrelease> {\default@shape}}}%
476 <latexrelease> \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
477 <latexrelease> \else
478 <latexrelease> \font@info{Redeclaring font encoding #1}%
479 <latexrelease> \fi
480 <latexrelease> \global\@namedef{T@#1}{#2}%
481 <latexrelease> \global\@namedef{M@#1}{\default@M#3}%
482 <latexrelease> \xdef\LastDeclaredEncoding{#1}%
483 <latexrelease> }
484 <latexrelease> \let\UseRawInputEncoding@\undefined
485 <latexrelease> \let\DeclareFontEncoding@saved@\undefined
486 <latexrelease> \let\inputencodingname@\undefined
487 <latexrelease> \EndIncludeInRelease
```

```
488 {*2ekernel}
```

We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
489 \def\reserved@a#1#2{%
490 \@tempcnta#1\relax
491 \@tempcntb#2\relax
492 \reserved@b
493 }
494 \def\reserved@b{%
495 \ifnum\@tempcnta>\@tempcntb\else
496 \reserved@c\@tempcnta
497 \advance\@tempcnta\@ne
498 \expandafter\reserved@b
499 \fi
500 }
```

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.

```
501 \catcode`\ =10
502 \catcode`\#=6
503 \catcode`\$=3
504 \catcode`\%=14
505 \catcode`\&=4
506 \catcode`\\=0
507 \catcode`\^=7
508 \catcode`_=8
509 \catcode`\{=1
510 \catcode`\}=2
511 \catcode`\-=13
512 \catcode`\@=11
513 \catcode`\^^I=10
514 \catcode`\^^J=12
515 \catcode`\^^L=13
516 \catcode`\^^M=5
```

Set the ‘other’ catcodes.

```
517 \def\reserved@c#1{\catcode#1=12\relax}
518 \reserved@cf`\'!
519 \reserved@cf`\""
520 \reserved@cf`\'?\}
521 \reserved@cf`\'[{}]
522 \reserved@cf`\'\]}
523 \reserved@cf`\'\}
524 \reserved@cf`\'\|}
```

Set the ‘letter’ catcodes.

```
525 \def\reserved@c#1{\catcode#1=11\relax}
526 \reserved@af`\'A\}\{\`Z\}
527 \reserved@af`\'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.

```

528 \ifnum 0%
529 \ifx\Umathcode\@undefined\else 1\fi
530 \ifx\XeTeXmathcode\@undefined\else 1\fi
531 >\z@
532 \else
533 \def\reserved@c#1{%
534 \count@=#1\advance\count@ by -"20
535 \uccode#1=\count@
536 \lccode#1=#1
537 }
538 \reserved@a{\a}{\z}
539 \reserved@a{"A0}{ "BC}
540 \reserved@a{"E0}{ "FF}

```

The upper case characters need their \uccode and \lccode values set, and their \sfcode set to 999.

```

541 \def\reserved@c#1{%
542 \count@=#1\advance\count@ by "20
543 \uccode#1=#1
544 \lccode#1=\count@
545 \sfcode#1=999
546 }
547 \reserved@a{\A}{\Z}
548 \reserved@a{"80}{ "9C}
549 \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.

```

550 \uccode`^Y='I % dotless i
551 \lccode`^Y='^Y % dotless i
552 \uccode`^Z='J % dotless j, ae in OT1
553 \lccode`^Z='^Z % dotless j, ae in OT1
554 \lccode`^9d='i % dotted I
555 \uccode`^9d='^9d % dotted I
556 \lccode`^9e='^9e % d-bar
557 \uccode`^9e='^d0 % d-bar

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

558 \lccode`^^[='^^[% oe in OT1
559 \fi % End of reset block for 8-bit engines

```

\BCPdata A stub for use by babel, polyglossia, etc.

```

560 \ExplSyntaxOn
561 \newcommand*\BCPdata[1]{
562 \str_case:nn {#1}
563 {
564 { language } { en }

```

```

565 { region } { US }
566 { script } { Latn }
567 { tag } { en-US }
568 }
569 }
570 \ExplSyntaxOff

```

(End of definition for \BCPdata.)

```

\MakeUppercase
\MakeLowercase
\MakeTitlecase
\NoCaseChange
\AddToNoCaseChangeList
 \CaseSwitch
\DeclareCaseChangeEquivalent
 \DeclareLowercaseMapping
\DeclareTitlecaseMapping
\DeclareUppercaseMapping
 \@uclclist

```

And whilst we're doing things with uc/lc tables, here are two commands to upper- and lower-case a string.

Wrappers around the L3 case changing functions. \protected to make them mostly safe as replacements for uppercase and \lowercase.

In

```
\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{\MakeUppercase Table of Contents}
 {\MakeUppercase Table of Contents}
```

In order to get round this, we redefine \MakeUppercase and \MakeLowercase to grab their argument and brace it.

Earlier versions needed to process \@uclclist in an \edef to handle legacy input encodings, but recent (2022) expl3 versions handle non-UTF8 text natively so we simply call the \text\_...\_case:n functions.

```

571 \ExplSyntaxOn
572 \keys_define:nn { __kernel }
573 {
574 lang .str_set:N = \reserved@a ,
575 locale .str_set:N = \reserved@a
576 }
577 \cs_new_protected:Npn \@@text@case@aux #1#2#3
578 {
579 \cs_set_nopar:Npn \reserved@a { }
580 \tl_if_blank:nTF {#2}
581 {
582 \str_set:Nx \reserved@a
583 { \BCPdata { casing } }
584 \str_if_empty:NT \reserved@a
585 {
586 \str_set:Nx \reserved@a
587 { \BCPdata { language } }
588 }
589 }
590 { \keys_set:nn { __kernel } {#2} }
591 \use:c { text_ #1 case:Vn } \reserved@a {#3}
592 }

```

The odd use of *three* spaces here is needed as \tcmd uses the name with one and two spaces to give a ‘friendly’ error message for a runaway argument: that means we can’t use it here.

```

593 \exp_args_generate:n { cnx }
594 \cs_set_protected:Npn \reserved@a #1
595 {
596 \cs_generate_variant:cn { text_ \str_lowercase:n {#1} case:nn } { V }
597 \ExpandArgs { cnx } \NewExpandableDocumentCommand
598 { Make#1case }
599 { O{} +m }
600 { \exp_not:c { Make#1case \c_space_tl \c_space_tl \c_space_tl } [####1] {####2} }
601 }
602 \reserved@a { Upper }
603 \reserved@a { Lower }
604 \reserved@a { Title }

```

Currently, babel uses the equivalence of \oe and \OE to force casing of some material, most notably in \today. To enable that to work, we have to set those commands equal even though the current case changing code does not work using this approach.

```

605 \cs_new_protected:cpn { MakeLowercase \c_space_tl \c_space_tl \c_space_tl } [#1] #2
606 {
607 \let \OE \oe
608 \@@text@case@aux { lower } {#1} {#2}
609 }
610 \cs_new_protected:cpn { MakeUppercase \c_space_tl \c_space_tl \c_space_tl } [#1] #2
611 {
612 \let \oe \OE
613 \@@text@case@aux { upper } {#1} {#2}
614 }
615 \cs_new_protected:cpn { MakeTitlecase \c_space_tl \c_space_tl \c_space_tl } [#1] #2
616 {
617 \let \oe \OE
618 \@@text@case@aux { title } {#1} {#2}
619 }

\NoCaseChange protects its argument from the case change functions.
\AddToNoCaseChangeList Allows new commands to protect their arguments, eg
AddToNoCaseChangeList{\eqref} would protect the argument of \eqref in the same
way as the argument of \ref.

620 \cs_new_protected_nopar:Npn \AddToNoCaseChangeList
621 {\tl_put_right:Nn \l_text_case_exclude_arg_tl}
622 \AddToNoCaseChangeList{ \NoCaseChange }
623 \cs_new_protected:Npn \NoCaseChange #1 {#1}
624 \cs_new_eq:NN \CaseSwitch \text_case_switch:nnnn
625 \cs_new_eq:NN \DeclareCaseChangeEquivalent
626 \text_declare_case_equivalent:Nn
627 \NewDocumentCommand \DeclareLowercaseMapping { o m m }
628 {
629 \IfNoValueTF {#1}
630 { \text_declare_lowercase_mapping:nn }
631 { \text_declare_lowercase_mapping:nnn {#1} }
632 {#2} {#3}
633 }
634 \NewDocumentCommand \DeclareTitlecaseMapping { o m m }

```

```

635 {
636 \IfNoValueTF {#1}
637 { \text_declare_titlecase_mapping:nn }
638 { \text_declare_titlecase_mapping:nnn {#1} }
639 {#2} {#3}
640 }
641 \NewDocumentCommand \DeclareUppercaseMapping { o m m }
642 {
643 \IfNoValueTF {#1}
644 { \text_declare_uppercase_mapping:nn }
645 { \text_declare_uppercase_mapping:nnn {#1} }
646 {#2} {#3}
647 }
648 \ExplSyntaxOff

649 \def\@uclclist{\oe\OE\o\O\ae\AE
650 \dh\DH\dj\DJ\l\L\ng\NG\ss\SS\ij\IJ\th\TH}

```

(End of definition for `\MakeUppercase` and others.)

## 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.

```

651 \% \IfFileExists{ltpatch.ltx}
652 % {\typeout{=====
653 % Applying patch file ltpatch.ltx^^J%
654 % =====}
655 % \def\fmtversion@topatch{unknown}
656 % \input{ltpatch.ltx}
657 % \ifx\fmtversion\fmtversion@topatch
658 % \ifx\patch@level\@undefined
659 % \typeout{^^J^^J^^J%
660 % !!!!!!!}
661 % !! Patch file 'ltpatch.ltx' not suitable for this^^J%
662 % !! version of LaTeX.^^J^^J%
663 % !! Please check if initex found an old patch file:^^J%
664 % !! --- if so, rename it or delete it, and redo the^^J%
665 % !! initex run.^^J%
666 % !!!!!!!}
667 % \batchmode \@@end
668 % \else

```

The code below adds the ‘patch level’ string to the first `\typeout` in the startup banner.

```

669 % \def\fmtversion@topatch{0}%
670 % \ifx\fmtversion@topatch\patch@level\else
671 % \def\reserved@a\typeout##1##2\reserved@a{%
672 % \typeout{##1 patch level \patch@level##2}
673 % \everyjob\expandafter\expandafter\expandafter{%
674 % \expandafter\reserved@a\the\everyjob\reserved@a}
675 % \let\reserved@a\relax
676 % \the\everyjob
677 % \fi

```

## 1.9 Freeing Memory

- `\reserved@a` And just to make sure nobody relies on those definitions of `\reserved@b` and friends.  
`\reserved@b` These macros are reserved for use in the kernel. *Do not use them as general scratch macros.*

```
692 \let\reserved@a=\@filelist
693 \let\reserved@b=\@undefined
694 \let\reserved@c=\@undefined
695 \let\reserved@d=\@undefined
696 \let\reserved@e=\@undefined
697 \let\reserved@f=\@undefined
```

(End of definition for \reserved@a and \reserved@b.)

\toks

```
698 \toks0{
699 \toks2{
700 \toks4{
701 \toks6{
702 \toks8{
```

(End of definition for \toks.)

- \errhelp Empty the error help message, which may have some rubbish:

703 \errhelp{}

(End of definition for \errhelp.)

## 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`.

```
704 \def\@providesfile#1[#2]{%
705 \wlog{File: #1 #2}%
706 \expandafter\xdef\csname ver@#1\endcsname{#2}%
707 }
```

(End of definition for \@providesfile.)

\@filelist Reset \@filelist so files input while making the format are not listed. The list built up so far may take up a lot of memory and so it is moved to \reserved@a where it will be overwritten as soon as almost any L<sup>A</sup>T<sub>E</sub>X 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.

```
708 \let\@filelist\@gobble
709 \def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}%
```

(End of definition for \@filelist and \@addtofilelist.)

## 1.11 Preparation for supporting PDF in backends

At the current point in time, basic support for PDF in backends is not part of L<sup>A</sup>T<sub>E</sub>X core; it is provided by external packages. At some time in the future that work will be placed into the kernel but for now it is separate and has to be explicitly loaded in the document.

In that code there is a command \IfPDFManagementActiveTF which can be used by packages in order to execute different code depending on the whether this basic backend support is loaded.

To make this also work properly when this external package is not loaded at all, we here add this command already in the kernel (with a trivial definition); thus any package can query this loading state in all circumstances. Once this basic PDF backend support gets moved to the kernel, this definition will vanish again from here or, rather, it will be replaced by a real test.

\IfPDFManagementActiveTF So long as the code for the basic backend support for PDF is not loaded, the test that is implicit here will always return the false branch. Once this code is loaded, this definition will get replaced by a real test (as it is then possible that the management code is either activated or not activated).

```
710 \let \IfPDFManagementActiveTF \@secondoftwo
```

(End of definition for \IfPDFManagementActiveTF.)

## 1.12 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 ...

## 1.13 Some last minute initializations ...

Load the first aid set of definitions for external packages that await updates.

```
711 \Qinput{latex2e-first-aid-for-external-files.ltx}
```

## 1.14 Dumping the format

Finally we make @ into a letter, ensure the format will be in the ‘normal’ error mode, and dump everything into the format file.

```
712 \makeatother
713 \errorstopmode
714 \dump
715 {/2ekernel}
```

# Change History

|                                                                                                                                                                                                                                      |     |                                                         |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------|-----|
| 1985-11-04 ltmath.dtx LaTeX2.09                                                                                                                                                                                                      |     | \mathversion: Test if version defined<br>added. . . . . | 534 |
| General: produce warning message if<br>line extends into margin. Doesn't<br>warn about formula overprinting<br>equation number. . . . .                                                                                              | 795 |                                                         |     |
| 1989-04-10 ltfssbas.dtx v1.0a                                                                                                                                                                                                        |     |                                                         |     |
| General: Starting with version<br>numbers! \ifmmode added in<br>\math@group . . . . .                                                                                                                                                | 523 |                                                         |     |
| 1989-04-10 ltfssbas.dtx v1.0b                                                                                                                                                                                                        |     |                                                         |     |
| General: \preload@sizes added. . . . .                                                                                                                                                                                               | 523 |                                                         |     |
| \wrong@fontshape changed to<br>define substitution font/shape<br>macro. . . . .                                                                                                                                                      | 523 |                                                         |     |
| 1989-04-10 ltfssini.dtx v1.0a                                                                                                                                                                                                        |     |                                                         |     |
| General: Starting with version<br>numbers \newif for \@tempswa<br>added since this switch is unknown<br>at the time when this file is read<br>in. (latex.tex is loaded later.)<br>\math@famname changed to<br>\math@version. . . . . | 634 |                                                         |     |
| 1989-04-14 ltfssbas.dtx v1.0c                                                                                                                                                                                                        |     |                                                         |     |
| General: More documentation added. . . . .                                                                                                                                                                                           | 523 |                                                         |     |
| 1989-04-15 ltfssini.dtx v1.0b                                                                                                                                                                                                        |     |                                                         |     |
| General: \mathfontset renamed to<br>\mathversion. . . . .                                                                                                                                                                            | 634 |                                                         |     |
| 1989-04-19 ltfssbas.dtx v1.0d                                                                                                                                                                                                        |     |                                                         |     |
| General: Even more doc. . . . .                                                                                                                                                                                                      | 523 |                                                         |     |
| 1989-04-21 ltfssbas.dtx v1.0e                                                                                                                                                                                                        |     |                                                         |     |
| General: Documentation is fun!<br>Parameters of<br>\define@mathalphabet changed. . . . .                                                                                                                                             | 523 |                                                         |     |
| 1989-04-21 ltfssini.dtx v1.0c                                                                                                                                                                                                        |     |                                                         |     |
| General: Changed to conform to<br>fam.tex. . . . .                                                                                                                                                                                   | 634 |                                                         |     |
| 1989-04-23 ltfssbas.dtx v1.0f                                                                                                                                                                                                        |     |                                                         |     |
| General: % in \getanddefinefonts<br>added. . . . .                                                                                                                                                                                   | 523 |                                                         |     |
| 1989-04-26 ltfssini.dtx v1.0d                                                                                                                                                                                                        |     |                                                         |     |
| General: \xpt added. . . . .                                                                                                                                                                                                         | 634 |                                                         |     |
| 1989-04-27 ltfssbas.dtx v1.0g                                                                                                                                                                                                        |     |                                                         |     |
| General: Documentation revised. . . . .                                                                                                                                                                                              | 523 |                                                         |     |
| 1989-04-27 ltfssini.dtx v1.0e                                                                                                                                                                                                        |     |                                                         |     |
| General: Definitions of L <sup>A</sup> T <sub>E</sub> X symbols<br>corrected. . . . .                                                                                                                                                | 634 |                                                         |     |
| 1989-04-29 ltfssbas.dtx v1.0h                                                                                                                                                                                                        |     |                                                         |     |
| General: Documented problem with<br>\halign, and \noalign . . . . .                                                                                                                                                                  | 523 |                                                         |     |
|                                                                                                                                                                                                                                      |     | \mathversion: Test if version defined<br>added. . . . . | 534 |
| 1989-04-29 ltfssbas.dtx v1.0i                                                                                                                                                                                                        |     |                                                         |     |
| General: Removed the \halign<br>\noalign correction (wasn't<br>bugfree) . . . . .                                                                                                                                                    | 523 |                                                         |     |
| 1989-04-29 ltfssini.dtx v1.0f                                                                                                                                                                                                        |     |                                                         |     |
| General: Corrections to L <sup>A</sup> T <sub>E</sub> X tabular<br>env. added. . . . .                                                                                                                                               | 634 |                                                         |     |
| 1989-05-01 ltfssbas.dtx v1.0j                                                                                                                                                                                                        |     |                                                         |     |
| General: Default for<br>\baselinestretch added. . . . .                                                                                                                                                                              | 523 |                                                         |     |
| 1989-05-22 ltfssbas.dtx v1.0k                                                                                                                                                                                                        |     |                                                         |     |
| General: Lines longer than 72<br>characters folded. . . . .                                                                                                                                                                          | 523 |                                                         |     |
| 1989-05-22 ltfssini.dtx v1.0g                                                                                                                                                                                                        |     |                                                         |     |
| General: Lines shortened to 72<br>characters . . . . .                                                                                                                                                                               | 634 |                                                         |     |
| 1989-09-14 ltfssbas.dtx v1.0m                                                                                                                                                                                                        |     |                                                         |     |
| General: Global replacement: \group<br>to \mathgroup . . . . .                                                                                                                                                                       | 523 |                                                         |     |
| \mathversion: Corrected typo:<br>\endcsname to \endcsmame. . . . .                                                                                                                                                                   | 534 |                                                         |     |
| 1989-11-07 ltfssini.dtx v1.0i                                                                                                                                                                                                        |     |                                                         |     |
| General: All family, series, and shape<br>names abbreviated. . . . .                                                                                                                                                                 | 634 |                                                         |     |
| 1989-11-08 ltfssbas.dtx v1.0o                                                                                                                                                                                                        |     |                                                         |     |
| General: First parameter of<br>\define@mathalphabet and<br>\define@mathgroup changed from<br>string to control sequence. . . . .                                                                                                     | 523 |                                                         |     |
| 1989-11-14 ltfssbas.dtx v1.0p                                                                                                                                                                                                        |     |                                                         |     |
| \mathversion: Math version prefix<br>'mv@' added. . . . .                                                                                                                                                                            | 534 |                                                         |     |
| 1989-11-19 ltfssbas.dtx v1.0q                                                                                                                                                                                                        |     |                                                         |     |
| \define@newfont: Group added. . . . .                                                                                                                                                                                                | 537 |                                                         |     |
| \wrong@fontshape: Instead of calling<br>\family\default@family, etc. we<br>directly set \f@family, etc. . . . .                                                                                                                      | 541 |                                                         |     |
| 1989-11-22 ltfssbas.dtx v1.0r                                                                                                                                                                                                        |     |                                                         |     |
| \mathversion: \def → \edef for<br>\mathversion. . . . .                                                                                                                                                                              | 534 |                                                         |     |
| 1989-11-25 ltfssbas.dtx v1.0s                                                                                                                                                                                                        |     |                                                         |     |
| General: All \edef\font@name<br>changed to \xdef\font@name.<br>Necessary after introduction of<br>\begingroup/\endgroup in v1.0q. . . . .                                                                                            | 523 |                                                         |     |
| extra// → + in \extra@def. . . . .                                                                                                                                                                                                   | 523 |                                                         |     |

|                                                                                                                                                     |     |                                                                                                                                             |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 1989-11-26 ltfssbas.dtx v1.0t<br>\select@group: \bgroup/\egroup<br>changed to<br>\begingroup/\endgroup to avoid<br>empty Ord atom on math list. . . | 544 | 1990-01-25 ltfssini.dtx v1.1e<br>\nfss@text: Macro added. . . . .                                                                           | 657 |
| 1989-12-02 ltfssini.dtx v1.1b<br>General: \rmmath renamed to<br>\mathrm . . . . .                                                                   | 634 | 1990-01-27 ltfssbas.dtx v1.2d<br>\DeclarePreloadSizes: Font identifier<br>set to \relax. . . . .                                            | 529 |
| 1989-12-03 ltfssini.dtx v1.1c<br>General: Some internal macros<br>renamed to make them<br>inaccessible. . . . .                                     | 634 | 1990-01-28 ltfssbas.dtx v1.2e<br>\mathgroup: \newfam let to<br>\new@mathgroup. . . . .                                                      | 523 |
| 1989-12-05 ltfssbas.dtx v1.0u<br>\addto@hook: \addto@hook added. . .                                                                                | 549 | 1990-01-28 ltfssbas.dtx v1.2f<br>\define@newfont: Added call to<br>\curr@fontshape macro to allow<br>substitution. . . . .                  | 537 |
| 1989-12-05 lfsstrc.dtx v1.0u fam.dtx<br>\every@math@size: Hook \every@size<br>added. . . . .                                                        | 581 | \wrong@fontshape: Warning message<br>slightly changed. . . . .                                                                              | 541 |
| 1989-12-13 lfsstrc.dtx v1.0f<br>\use@mathgroup: \expandafter added<br>before final \fi. . . . .                                                     | 584 | 1990-01-28 ltfssini.dtx v1.2b<br>\em: Call to \nomath added. . . . .                                                                        | 654 |
| 1989-12-16 ltfssbas.dtx v1.1a<br>\select@group: \relax in front<br>added. . . . .                                                                   | 544 | 1990-02-08 ltfssini.dtx v1.1g<br>General: Protected the commands<br>\fam, \series, \shape, \size,<br>\selectfont, and \mathversion. . . . . | 634 |
| Now four arguments. . . . .                                                                                                                         | 544 | 1990-02-16 ltfssbas.dtx v1.2g<br>General: Support for changes of<br>\baselineskip without changing<br>the size. . . . .                     | 523 |
| Redefinition of alphabet now<br>simpler. . . . .                                                                                                    | 544 | \mathversion: \nomath added. . .                                                                                                            | 534 |
| Usage of '=' macro added. . . . .                                                                                                                   | 544 | 1990-02-18 lfsstrc.dtx v1.0j<br>\selectfont: Redefine unprotected<br>version \p@selectfont instead of<br>\selectfont. . . . .               | 576 |
| 1989-12-16 lfsstrc.dtx v1.1a<br>\selectfont: Changed order of calls. . .                                                                            | 576 | 1990-03-14 lfsstrc.dtx v1.0k<br>General: Added code for TeX3. . . . .                                                                       | 572 |
| \use@mathgroup: Redefinition of<br>alphabet now simpler. . . . .                                                                                    | 583 | \extract@font: Added code for<br>TeX3. . . . .                                                                                              | 575 |
| Usage of '=' macro added. . . . .                                                                                                                   | 583 | 1990-03-30 ltfssbas.dtx v1.2h<br>\math@egroup: Changed to have one<br>arg. . . . .                                                          | 546 |
| 1990-01-18 lfsstrc.dtx v1.0h<br>General: \tracingfonts meaning<br>changed. . . . .                                                                  | 572 | 1990-03-30 lfsstrc.dtx v1.2h<br>\use@mathgroup: Third argument<br>removed (see \math@egroup). . .                                           | 583 |
| 1990-01-20 ltfssbas.dtx v1.2a<br>\math@bgroup: Def. placed in this<br>file. . . . .                                                                 | 546 | 1990-04-01 ltfssbas.dtx v1.2i<br>General: Code added from<br>tracefn.tdx. . . . .                                                           | 523 |
| \math@egroup: Def. placed in this<br>file. . . . .                                                                                                  | 546 | Support for TeX3. . . . .                                                                                                                   | 523 |
| \select@group: Def for alph id<br>changed. . . . .                                                                                                  | 544 | 1990-04-01 lfsstrc.dtx v1.0l<br>General: Part of code moved to<br>fam.dtx. . . . .                                                          | 572 |
| 1990-01-21 ltfssbas.dtx v1.2b<br>\select@group: Code moved to<br>\use@mathgroup. . . . .                                                            | 544 | \tracingfonts: Check if<br>\tracingfonts already defined. .                                                                                 | 573 |
| 1990-01-21 lfsstrc.dtx v1.2b<br>\use@mathgroup: Macro added to<br>allow cleaner interface. . . . .                                                  | 583 | 1990-04-01 lfsstrc.dtx v1.0o<br>\tracingfonts: Check if<br>\tracingfonts defined removed<br>again. . . . .                                  | 573 |
| 1990-01-23 ltfssbas.dtx v1.2c<br>General: \no@version@warning<br>renamed to \no@alphabet@error. . .                                                 | 523 |                                                                                                                                             |     |
| Macro \no@alphabet@help added . .                                                                                                                   | 523 |                                                                                                                                             |     |
| \no@alphabet@error: Changed to<br>error call . . . . .                                                                                              | 523 |                                                                                                                                             |     |

|                                                                                                             |     |                                                                                                                                               |     |
|-------------------------------------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 1990-04-02 ltfssini.dtx v1.1i                                                                               |     | 1991-08-14 ltpictur.dtx LaTeX2.09                                                                                                             |     |
| General: \input of files now handled<br>by docstrip. . . . .                                                | 634 | General: (RmS) inserted extra braces<br>around entry for NFSS . . . . .                                                                       | 867 |
| 1990-04-05 lfsstrc.dtx v1.0m                                                                                |     | 1991-08-14 ltthm.dtx LaTeX2.09                                                                                                                |     |
| \selectfont: Call \tracingon only if<br>\tracingfonts greater than 3. . . . .                               | 576 | \endtheorem: Moved \itshape after<br>\item to make it work with NFSS . . . . .                                                                | 897 |
| 1990-05-05 lfsstrc.dtx v1.0n                                                                                |     | 1991-08-26 ltfssini.dtx v1.1n                                                                                                                 |     |
| \selectfont: \tracingon with new<br>syntax. . . . .                                                         | 576 | \reset@font: Macro introduced . . . . .                                                                                                       | 657 |
| 1990-06-23 ltfssini.dtx v1.1k                                                                               |     | 1991-08-26 ltmiscen.dtx LaTeX2.09                                                                                                             |     |
| \nfss@text: Changed to \mbox. . . . .                                                                       | 657 | \overline: \@par added . . . . .                                                                                                              | 778 |
| 1990-06-24 ltfssbas.dtx v1.2j                                                                               |     | 1991-08-26 ltpictur.dtx LaTeX2.09                                                                                                             |     |
| \DeclarePreloadSizes: Missing<br>percent added. . . . .                                                     | 529 | \endpicture: (RmS & FMi) extra<br>boxing level around \picbox to<br>guard against unboxing in math<br>mode (proposed by John Hobby) . . . . . | 865 |
| 1990-06-24 lfsstrc.dtx v1.0o                                                                                |     | 1991-08-26 ltplain.dtx LaTeX209                                                                                                               |     |
| \baselinestretch: Moved to<br>tracefntr.dtx. . . . .                                                        | 581 | \tracingall: Added<br>\errorcontextlines=\maxdimen,<br>suggested by J. Schrod . . . . .                                                       | 33  |
| \getanddefine@fonts: \Adding<br>tracing code. . . . .                                                       | 585 | 1991-09-29 ltboxes.dtx LaTeX2.09                                                                                                              |     |
| \Macro moved from fam.dtx. . . . .                                                                          | 584 | \cmpfootnotetext: (RmS) added<br>\reset@font . . . . .                                                                                        | 832 |
| Adding debug code. . . . .                                                                                  | 585 | 1991-09-29 ltfloat.dtx LaTeX2.09                                                                                                              |     |
| \use@mathgroup: Tracing code added. . . . .                                                                 | 584 | \footnotetext: (RmS) added<br>\reset@font . . . . .                                                                                           | 930 |
| 1990-06-30 ltfssbas.dtx v1.2l                                                                               |     | 1991-09-29 ltmath.dtx LaTeX2.09                                                                                                               |     |
| \showhyphens: Macro added. . . . .                                                                          | 547 | \eqnnum: RmS: \reset@font added. . . . .                                                                                                      | 794 |
| 1990-06-30 lfsstrc.dtx v1.0p                                                                                |     | 1991-09-29 ltsect.dtx LaTeX2.09                                                                                                               |     |
| \use@mathgroup: Added \relax after<br>math group number. . . . .                                            | 584 | \dottedtocline: (RmS) added<br>\reset@font for page number . . . . .                                                                          | 909 |
| 1990-07-07 lfsstrc.dtx v1.0q                                                                                |     | 1991-10-17 ltcntrl.dtx LaTeX209                                                                                                               |     |
| \getanddefine@fonts: Group number<br>added to tracing. . . . .                                              | 585 | \ctfor: (Rms) \xdef replaced by \def<br>(See FMi's array.doc) . . . . .                                                                       | 387 |
| \math@egroup: Tracing code added. . . . .                                                                   | 584 | 1991-10-25 ltbibl.dtx LaTeX2.09                                                                                                               |     |
| \use@mathgroup: Group number<br>added to tracing. . . . .                                                   | 584 | \citex: added \reset@font,<br>suggested by Bernd Raichle. . . . .                                                                             | 939 |
| 1990-08-27 lfsstrc.dtx 1.0r                                                                                 |     | 1991-11-01 ltfloat.dtx LaTeX2.09                                                                                                              |     |
| \type@restoreinfo: Some extra<br>tracing info. . . . .                                                      | 580 | \footnote: (RmS) Added<br>\let\protect\noexpand in<br>\footnote, \footnotemark, and<br>\footnotetext, since \xdef is<br>used . . . . .        | 930 |
| 1990-08-27 lfsstrc.dtx v1.0r                                                                                |     | 1991-11-04 ltlists.dtx LaTeX2.09                                                                                                              |     |
| \getanddefine@fonts: Correcting<br>missing name after \tracingon. . . . .                                   | 585 | \makelabel: (RmS) added default<br>definition for \makelabel, to<br>produce an error message. . . . .                                         | 815 |
| 1991-03-28 ltfssini.dtx v1.1m                                                                               |     | 1991-11-04 ltplain.dtx RmS                                                                                                                    |     |
| \copyright: Extra braces added. . . . .                                                                     | 657 | General: Removed \itemitem since<br>never needed/useful with L <sup>A</sup> T <sub>E</sub> X. . . . .                                         | 31  |
| 1991-03-30 ltfssini.dtx v1.2g                                                                               |     | 1991-11-06 ltbibl.dtx LaTeX2.09                                                                                                               |     |
| \newfont: Definition added. . . . .                                                                         | 656 | \citex: added code to remove a<br>leading blank . . . . .                                                                                     | 939 |
| \symbol: Definition added. . . . .                                                                          | 656 |                                                                                                                                               |     |
| 1991-07-24 ltmiscen.dtx LaTeX2.09                                                                           |     |                                                                                                                                               |     |
| \verbatim: Added<br>\penalty\interlinepenalty to<br>definition of \par so that<br>\samepage works . . . . . | 778 |                                                                                                                                               |     |
| 1991-08-14 ltmath.dtx LaTeX2.09                                                                             |     |                                                                                                                                               |     |
| \cases: (RmS) inserted extra braces<br>around entry for NFSS . . . . .                                      | 789 |                                                                                                                                               |     |

|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------------------------------------------------------------------|-----------------------------------------------------------------|-------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------|-----|--|
| 1991-11-13 | ltbibl.dtx   | LaTeX2.09                                                                                                                                        |     | avoid conflicts with other channels<br>allocated by \newread . . . . . | 82                                                              |             |           |                                                                                                                                            |     |  |
|            | \@bibitem:   | Changed counter enumi to<br>enumiv, as it says in the comment<br>above . . . . .                                                                 | 938 |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
| 1991-11-21 | ltfssini.dtx | v1.10                                                                                                                                            |     | \@xmpar: (RmS) added<br>\global\@ignorefalse . . . . .                 | 925                                                             |             |           |                                                                                                                                            |     |  |
|            | \reset@font: | Added extra braces for<br>robustness. . . . .                                                                                                    | 657 | \end@float: (RmS) changed<br>\@esphack to \@Espack . . . . .           | 919                                                             |             |           |                                                                                                                                            |     |  |
|            |              | Changed to protected version of<br>macro. . . . .                                                                                                | 657 |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
| 1991-11-22 | ltfloat.dtx  | LaTeX2.09                                                                                                                                        |     | \trivlist: RmS: added<br>\@nmbrlistfalse . . . . .                     | 810                                                             |             |           |                                                                                                                                            |     |  |
|            | \footnote:   | (RmS) Added<br>\let\protect\noexpand in<br>\xfootnote, \xfootnotemark,<br>and \xfootnotetext . . . . .                                           | 930 | 1992-03-18                                                             | ltmiscren.dtx                                                   | LaTeX2.09   |           | \begin: Changed \@ignoretrue to<br>\@ignorefalse (as documented)                                                                           | 769 |  |
| 1991-11-22 | ltlists.dtx  | LaTeX2.09                                                                                                                                        |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            | \@item:      | (RmS) Changed second call to<br>\makelabel to<br>\unhbox\@tempboxa. Avoids<br>problems with side effects in<br>\makelabel and is more efficient. | 815 | 1992-03-21                                                             | ltfssini.dtx                                                    | v1.2d       |           | General: Renamed \text to<br>\nfss@text to make it internal.                                                                               | 634 |  |
| 1991-11-27 | ltfssbas.dtx | v1.3a                                                                                                                                            |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            | General:     | All \family, \shape etc.<br>renamed to \fontfamily etc. . . .                                                                                    | 523 | 1992-05-12                                                             | ltfssbas.dtx                                                    | v1.3c       |           | \extract@alph@from@version: Macro<br>added. . . . .                                                                                        | 545 |  |
| 1991-11-27 | ltfssini.dtx | v1.2a                                                                                                                                            |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            | General:     | All \family, \shape etc.<br>renamed to \fontfamily etc. . . .                                                                                    | 634 |                                                                        | \select@group: Added call to<br>\extract@alph@from@version. . . | 545         |           |                                                                                                                                            |     |  |
| 1992-01-06 | ltfssini.dtx | v1.2c                                                                                                                                            |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            | General:     | added slitex code . . . . .                                                                                                                      | 634 | 1992-07-26                                                             | ltfssbas.dtx                                                    | v1.9a       |           | \curr@fontshape: . . . . .                                                                                                                 | 536 |  |
| 1992-01-10 | ltbibl.dtx   | LaTeX2.09                                                                                                                                        |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            | \@bibitem:   | Changed \c@enumiv to<br>\value of \@listctr . . . . .                                                                                            | 938 |                                                                        | \DeclareFontShape: Introduced<br>\DeclareFontShape . . . . .    | 524         |           |                                                                                                                                            |     |  |
| 1992-01-10 | ltmath.dtx   | LaTeX2.09                                                                                                                                        |     |                                                                        | \define@newfont: . . . . .                                      | 537         |           |                                                                                                                                            |     |  |
|            | equation:    | RmS: put \hbox around<br>\eqnnum to typeset the equation<br>number in text mode (as in the<br>eqnarray env.) . . . . .                           | 794 |                                                                        | \math@fonts: . . . . .                                          | 544         |           |                                                                                                                                            |     |  |
| 1992-01-10 | ltthm.dtx    | LaTeX2.09                                                                                                                                        |     |                                                                        | \select@group: . . . . .                                        | 544, 545    |           |                                                                                                                                            |     |  |
|            | \@othm:      | (RmS) Check for existence of<br>theorem environment . . . . .                                                                                    | 896 |                                                                        | \split@name: Added splitting into<br>\f@encoding. . . . .       | 536         |           |                                                                                                                                            |     |  |
| 1992-01-14 | ltbibl.dtx   | LaTeX2.09                                                                                                                                        |     |                                                                        | \wrong@fontshape: . . . . .                                     | 541, 542    |           |                                                                                                                                            |     |  |
|            | \@biblabel:  | removed \hfill . . . . .                                                                                                                         | 941 | 1992-07-26                                                             | ltfsstrc.dtx                                                    | v2.0b       |           |                                                                                                                                            |     |  |
| 1992-01-14 | ltsect.dtx   | 0.0                                                                                                                                              |     |                                                                        | \s@fct@: . . . . .                                              | 593         |           |                                                                                                                                            |     |  |
|            | \@starttoc:  | (RmS) added \immediate<br>to \openout as all \write<br>commands are also executed<br>\immediate . . . . .                                        | 907 |                                                                        | \s@fct@sub: documentation fixes . .                             | 594         |           |                                                                                                                                            |     |  |
| 1992-02-26 | ltbibl.dtx   | LaTeX2.09                                                                                                                                        |     |                                                                        | \selectfont: . . . . .                                          | 577         |           |                                                                                                                                            |     |  |
|            | \@lbibitem:  | Added \hfill to restore<br>left-alignment of bibliography<br>labels in alpha style . . . . .                                                     | 938 |                                                                        | \try@simple@size: . . . . .                                     | 587         |           |                                                                                                                                            |     |  |
| 1992-03-18 | ltdefns.dtx  | LaTeX209                                                                                                                                         |     |                                                                        | \try@size@range: . . . . .                                      | 591         |           |                                                                                                                                            |     |  |
|            | General:     | (RMS) changed input<br>channel from 0 to \inputcheck to                                                                                          |     |                                                                        | \use@mathgroup: . . . . .                                       | 584         |           |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        | 1992-08-14                                                      | ltbibl.dtx  | LaTeX2.09 |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           | \@citex: added missing argument<br>braces around \hbox, found by Ed<br>Sznyter . . . . .                                                   | 939 |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        | 1992-08-14                                                      | ltboxes.dtx | LaTeX209  |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           | \endminipage: (RmS) replaced<br>\vskip-\lastskip by \unskip<br>(proposed by FMI) . . . . .                                                 | 832 |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        | 1992-08-17                                                      | ltbibl.dtx  | LaTeX2.09 |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           | \@citex: simplified code for removing<br>leading blanks in citation key<br>(proposed by Frank Jensen and<br>Kresten Krab Thorup) . . . . . | 939 |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        | 1992-08-19                                                      | ltsect.dtx  | 0.0       |                                                                                                                                            |     |  |
|            |              |                                                                                                                                                  |     |                                                                        |                                                                 |             |           | \@xsect: (RmS) corrected bug: stretch<br>and shrink in argument to \hskip                                                                  |     |  |

|                                                                                                                                     |     |                                                                                                                                                    |     |
|-------------------------------------------------------------------------------------------------------------------------------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| previously not negated . . . . .                                                                                                    | 903 | \footnote: (RmS) Changed all to<br>‘def’protect’noexpand’protect’noexpand<br>. . . . .                                                             | 930 |
| 1992-08-19 ltthm.dtx LaTeX2.09                                                                                                      |     |                                                                                                                                                    |     |
| \@othm: (RmS) Changed error message<br>to complain about undefined<br>counter . . . . .                                             | 896 | \hexnumber@: Make it accept<br>counters. . . . .                                                                                                   | 657 |
| 1992-08-20 ltfssini.dtx v1.4b                                                                                                       |     |                                                                                                                                                    |     |
| \@setsizE: Added \@currsize. . . . .                                                                                                | 656 | 1993-03-08 preload.dtx v2.0b<br>General: Added 12pt preloads . . . . .                                                                             | 683 |
| 1992-08-24 ltdefns.dtx LaTeX209                                                                                                     |     |                                                                                                                                                    |     |
| \@ifnextchar: (Rms) \@ifnextchar<br>didn't work if its first argument<br>was an equal sign. . . . .                                 | 107 | 1993-03-18 ltfssbas.dtx v2.0c<br>General: Changed all \@tempdima in<br>\@tempdimb to avoid killing<br>\numberline . . . . .                        | 523 |
| 1992-08-24 ltmiscen.dtx LaTeX2.09                                                                                                   |     |                                                                                                                                                    |     |
| \begin: Added code to \begin to<br>remember line number. Used by<br>\@badend to display position of<br>non-matching \begin. . . . . | 769 | 1993-03-18 lfsstrc.dtx v2.1b<br>General: Changed all \@tempdima in<br>\@tempdimb to avoid killing<br>\numberline . . . . .                         | 572 |
| \verb: Changed \verb and \@sverb to<br>work correctly in math mode . . . . .                                                        | 783 | Changed all \@tempdimb in<br>\@tempdimx to avoid killing<br>\numberline . . . . .                                                                  | 572 |
| 1992-08-25 ltsect.dtx LaTeX2.09                                                                                                     |     |                                                                                                                                                    |     |
| \@sect: (FMi) replaced explicit setting<br>of \@svsec by call to<br>\@seccntformat . . . . .                                        | 902 | 1993-03-18 lfsstrc.dtx v2.1c<br>\DeclareSizeFunction: Added all<br>args to avoid blanks problems . . . . .                                         | 590 |
| 1992-09-18 ltlists.dtx LaTeX2.09                                                                                                    |     |                                                                                                                                                    |     |
| \item: (RmS) Added warning if \item<br>is used in math mode . . . . .                                                               | 813 | 1993-04-09 lterror.dtx v1.0e<br>\@latexerr: Mention The<br>Companion . . . . .                                                                     | 394 |
| 1992-09-18 ltab.dtx LaTeX2.09                                                                                                       |     |                                                                                                                                                    |     |
| \@array: Changed \par to \empty to<br>avoid starting new row e.g. after<br>\hline . . . . .                                         | 850 | 1993-04-11 lterror.dtx v1.0f<br>\@latexerr: Remove setting of<br>errorcontextlines . . . . .                                                       | 394 |
| 1992-09-19 lfsstrc.dtx v2.0c                                                                                                        |     |                                                                                                                                                    |     |
| \try@simple@size: . . . . .                                                                                                         | 587 | 1993-05-05 lftntcmd.dtx v2.0b<br>General: Removed all LaTeX related<br>cmds . . . . .                                                              | 686 |
| 1992-09-21 ltfssini.dtx v1.4d                                                                                                       |     |                                                                                                                                                    |     |
| \@not@math@alphabet: Macro defined. . . . .                                                                                         | 655 | 1993-05-16 ltfssbas.dtx v2.0e<br>\showhyphens: Use \reset@font . . . . .                                                                           | 547 |
| 1992-09-22 ltfssbas.dtx v1.91a                                                                                                      |     |                                                                                                                                                    |     |
| General: Introduced \tf@size for<br>math size. . . . .                                                                              | 523 | 1993-07-16 lfsstrc.dtx v2.1h<br>General: Changed layout of info<br>messages . . . . .                                                              | 572 |
| 1992-09-22 lfsstrc.dtx v2.1a                                                                                                        |     |                                                                                                                                                    |     |
| \getanddefine@fonts: Introduced<br>\@tf@size for math size. . . . .                                                                 | 585 | 1993-07-17 ltoutenc.dtx 1.0d<br>General: changed \catcoding @ . . . . .                                                                            | 469 |
| 1992-11-13 ltfssini.dtx v?                                                                                                          |     |                                                                                                                                                    |     |
| \hexnumber@: Made expandable. . . . .                                                                                               | 657 | 1993-08-03 ltmiscen.dtx LaTeX2.09<br>\enddocument: Changed redefinition of<br>\global to redefinition of<br>\@setckpt. . . . .                     | 761 |
| 1992-11-23 ltcounds.dtx LaTeX209                                                                                                    |     |                                                                                                                                                    |     |
| \stepcounter: Replaced {} in<br>\stepcounter by \begingroup<br>\endgroup to avoid adding an<br>empty ord in math mode . . . . .     | 514 | 1993-08-05 ltpictur.dtx LaTeX2.09<br>\circle: (RMS) Added error message<br>if \circle is used in math mode. . . . .                                | 887 |
| 1992-11-26 ltboxes.dtx LaTeX2.09                                                                                                    |     |                                                                                                                                                    |     |
| \@mpfootnotetext: (RmS) added<br>protection for \edef . . . . .                                                                     | 832 | 1993-08-05 ltsect.dtx LaTeX2.09<br>\@sect: (RmS) Made sure that<br>\protect works correctly in<br>expansion of \the counter . . . . .              | 902 |
| 1992-11-26 lfloat.dtx LaTeX2.09                                                                                                     |     |                                                                                                                                                    |     |
| \@footnotetext: (RmS) added<br>protection for \edef . . . . .                                                                       | 930 | 1993-08-05 ltspace.dtx LaTeX2e<br>\@hspacE: (RmS) Removed<br>superfluous \leavevmode in<br>\@hspacE and \@hspacEr, as<br>suggested by CAR. . . . . | 439 |

|                                                                                                                                              |                                    |     |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|-----|
| 1993-08-05 lttab.dtx latex2e                                                                                                                 | \everycr . . . . .                 | 867 |
| \tabular*: Replaced<br>\expandafter\def by \cnamedef.                                                                                        | 849                                |     |
| 1993-08-06 ltbibl.dtx LaTeX2.09                                                                                                              |                                    |     |
| \citet: Moved writing to .aux file in<br>loop over citation keys so that<br>leading blanks are removed there<br>as well. . . . .             | 939                                |     |
| 1993-08-13 ltoutenc.dtx 1.0f                                                                                                                 |                                    |     |
| General: Protected against active @<br>sign. . . . .                                                                                         | 469                                |     |
| 1993-08-13 preload.dtx v2.0c                                                                                                                 |                                    |     |
| General: Added \relax at end of font<br>names. . . . .                                                                                       | 684                                |     |
| 1993-08-16 ltoutenc.dtx 1.0g                                                                                                                 |                                    |     |
| General: Needs space after \string                                                                                                           | 469                                |     |
| 1993-08-18 ltfssdcl.dtx v2.0e                                                                                                                |                                    |     |
| \new@mathversion: Exchanged names<br>of encodings in warning message of<br>\SetSymbolFont. . . . .                                           | 614                                |     |
| 1993-09-02 lfsstrc.dtx v2.1i                                                                                                                 |                                    |     |
| General: Corrected name of sgen size<br>function. . . . .                                                                                    | 572                                |     |
| 1993-09-03 ltmiscen.dtx LaTeX2.09                                                                                                            |                                    |     |
| \verbatim@nolig@list: Replaced<br>\@noligs by extensible list . . .                                                                          | 783                                |     |
| 1993-09-07 ltmiscen.dtx LaTeX2.09                                                                                                            |                                    |     |
| \verb@balance@group: (RmS)<br>Changed definition of \verb so<br>that it detects a missing second<br>delimiter. . . . .                       | 783                                |     |
| 1993-09-08 ltmiscen.dtx LaTeX2.09                                                                                                            |                                    |     |
| \enddocument: Added warning in case<br>of undefined references. . . . .                                                                      | 761                                |     |
| 1993-09-15 ltfsbas.dtx v2.0g                                                                                                                 |                                    |     |
| \DeclareFontEncoding: Corrected:<br>\default@T to \default@M. . . .                                                                          | 527                                |     |
| 1993-09-15 lfsstrc.dtx v2.1j                                                                                                                 |                                    |     |
| General: Corrected spelling of<br>\noxpand. . . . .                                                                                          | 572                                |     |
| 1993-09-19 lterror.dtx LaTeX2.09                                                                                                             |                                    |     |
| \@invalidchar: (RmS) Error message<br>for invalid input characters. . . .                                                                    | 398                                |     |
| 1993-11-02 ltmath.dtx LaTeX2.09                                                                                                              |                                    |     |
| General: RmS: Corrected description<br>of \eqnse1, moved \eqnse1<br>accordingly and removed extra<br>\tabskip assignment. . . . .            | 795                                |     |
| 1993-11-03 ltmath.dtx LaTeX2e                                                                                                                |                                    |     |
| General: RmS: Initialized \everycr to<br>empty . . . . .                                                                                     | 795                                |     |
| 1993-11-03 ltpictur.dtx LaTeX2.09                                                                                                            |                                    |     |
| General: (RmS) changed \halign to<br>\ialign to initialize \tabskip and                                                                      |                                    |     |
|                                                                                                                                              | \everycr . . . . .                 | 867 |
| 1993-11-11 ltfssini.dtx v2.1a                                                                                                                | \normalfont: Macro added . . . . . | 657 |
| 1993-11-11 lfsstrc.dtx v2.2a                                                                                                                 |                                    |     |
| General: Option concept added for<br>LaTeX2e . . . . .                                                                                       | 572                                |     |
| 1993-11-14 ltclass.dtx v0.2a                                                                                                                 |                                    |     |
| \currext: Name changed from<br>\currextension . . . . .                                                                                      | 979                                |     |
| \resetoptions: macro added . . .                                                                                                             | 1007                               |     |
| \AtEndDocument: Included extension<br>in the generated macro name for<br>package and class hooks. . . . .                                    | 1007                               |     |
| \documentstyle: Added<br>\RequirePackage                                                                                                     |                                    |     |
| \@unuseoptionlist stuff. . . .                                                                                                               | 995                                |     |
| \load@onefilewithoptions: Moved<br>resetting of \default@ds, \ds@ and<br>\@declaredoptions here, from the<br>end of \ProcessOptions. . . . . | 1001                               |     |
| \NeedsTeXFormat: made more robust<br>for alternative syntax for other<br>formats. . . . .                                                    | 997                                |     |
| \ProcessOptions*: Optimize ‘empty<br>option’ code. . . . .                                                                                   | 992                                |     |
| Stop adding the global option list<br>inside class files. . . . .                                                                            | 992                                |     |
| 1993-11-14 ltdefns.dtx v0.2a                                                                                                                 | \g@addto@macro: Made global . . .  | 112 |
| 1993-11-15 ltclass.dtx v0.2b                                                                                                                 |                                    |     |
| \documentstyle: Modified to match<br>\ProcessOption*. . . . .                                                                                | 995                                |     |
| \ProcessOptions*: Star form added. .                                                                                                         | 992                                |     |
| 1993-11-17 ltclass.dtx v0.2c                                                                                                                 |                                    |     |
| \@fileswith@ptions: Macro added                                                                                                              | 1007                               |     |
| \@badrequireerror: Macro added                                                                                                               | 1009                               |     |
| \@twoloadclasserror: Macro added                                                                                                             | 1009                               |     |
| \CurrentOption: Name changed from<br>\@curroption . . . . .                                                                                  | 979                                |     |
| \DeclareOption*: Error checking<br>added . . . . .                                                                                           | 991                                |     |
| \load@onefilewithoptions: Added<br>trap for two \LoadClass<br>commands. . . . .                                                              | 1003                               |     |
| \NeedsTeXFormat: Name changed from<br>\NeedsFormat . . . . .                                                                                 | 997                                |     |
| \ProcessOptions*: restoring<br>\@fileswith@ptions added. . .                                                                                 | 992                                |     |
| 1993-11-18 ltclass.dtx v0.2d                                                                                                                 |                                    |     |
| \documentstyle: Modified<br>\RequirePackage stuff. . . . .                                                                                   | 995                                |     |
| \ExecuteOptions: Use<br>\CurrentOption not \reserved@a                                                                                       | 995                                |     |

|                                          |         |                                                               |         |
|------------------------------------------|---------|---------------------------------------------------------------|---------|
| \NeedsTeXFormat: \fmtname                | 997     | \newcommand: Macro reimplemented and extended . . . . .       | 83      |
| \fmtversion not @... . . . .             |         | and extended . . . . .                                        |         |
| 1993-11-21 ltfiles.dtx                   | LaTeX2e | \renewcommand: Macro reimplemented and extended . . . . .     | 86      |
| @missingfileerror: Stop infinite         |         | and extended . . . . .                                        |         |
| looping on @er@ext . . . . .             | 461     | \renewenvironment: Macro reimplemented and extended . . . . . | 87      |
| 1993-11-21 ltmiscen.dtx                  | v0.9a   | \two@digits: Macro added . . . . .                            | 80      |
| @verbatim: use \verb@font instead of \tt | 778     | 1993-11-23 ltoutput.dtx v0.1a                                 |         |
| @verb: Use \verb@font instead of \tt     | 783     | \paperheight: Register added . . . . .                        | 1099    |
| @verbatim@font: Macro added . . . . .    | 779     | \paperwidth: Register added . . . . .                         | 1099    |
| 1993-11-22 ltclass.dtx                   | v0.2f   | 1993-11-23 ltoutput.dtx v0.1c                                 |         |
| @fileswithoptions: Made the              |         | \enlargepage: Command added . . . . .                         | 1144    |
| default [] not [\@unknversion]           | 999     | \kludgeins: Insert added . . . . .                            | 1143    |
| @ifl@ter: Added //0 so parsing           |         | \makecol: Command changed . . . . .                           | 1111    |
| never produces a runaway                 |         | @specialoutput: Command                                       |         |
| argument . . . . .                       | 984     | changed . . . . .                                             | 1104    |
| General: \@unknversion removed           | 974     | \enlargethispage*: Commands                                   |         |
| \load@onefilewithoptions: Made the       |         | added . . . . .                                               | 1143    |
| initial version [] not                   |         | 1993-11-24 ltfnctcmd.dtx v2.1a                                |         |
| [\@unknversion] . . . . .                | 1001    | \maybe@ic@: Use \t@st@ic . . . . .                            | 691     |
| 1993-11-22 ltdefns.dtx                   | LaTeX2e | \t@st@ic: Macro added . . . . .                               | 692     |
| @minus: Macro added . . . . .            | 81      | 1993-11-24 ltfssini.dtx v2.1a                                 |         |
| @plus: Macro added . . . . .             | 81      | General: Removed \xpt stuff . . . . .                         | 657     |
| \CheckCommand: Macro added . . . . .     | 88      | 1993-11-24 ltlogos.dtx                                        | LaTeX2e |
| \providetcommand: Macro added . . . . .  | 88      | \LaTeX: Macro changed . . . . .                               | 441     |
| 1993-11-22 lterror.dtx                   | LaTeX2e | 1993-11-28 ltclass.dtx v0.2h                                  |         |
| @c@errorcontextlines: Macro added        | 394     | @twoclasseserror: Macro added . . . . .                       | 1009    |
| 1993-11-22 ltfiles.dtx                   | LaTeX2e | General: Assorted commands now in                             |         |
| @listfiles: Removed checking for         |         | the kernel removed . . . . .                                  | 978     |
| @unknversion . . . . .                   | 463     | Directory syntax checking moved to                            |         |
| 1993-11-22 ltlength.dtx                  | LaTeX2e | dircheck.dtx . . . . .                                        | 978     |
| @settodim: Macro added . . . . .         | 521     | Primitive filenames now terminated                            |         |
| @settopoint: Macro added . . . . .       | 522     | by space not \relax . . . . .                                 | 978     |
| @settodepth: Macro added . . . . .       | 521     | @endfilecontents: Don't globally                              |         |
| @settoheight: Macro added . . . . .      | 521     | allocate a write stream (always use                           |         |
| 1993-11-22 ltlogos.dtx                   | LaTeX2e | 15) . . . . .                                                 | 1010    |
| \LaTeXe: Macro added . . . . .           | 441     | 1993-11-28 ltfiles.dtx                                        | LaTeX2e |
| 1993-11-23 ltclass.dtx                   | v0.2g   | @missingfileerror: Use filename                               |         |
| @use@option: Name changed from           |         | parser from dircheck . . . . .                                | 461     |
| @executeoption . . . . .                 | 994     | 1993-11-29 ltoutput.dtx v1.0b                                 |         |
| General: Various macros now moved        |         | @makecol: \@makespecialcolbox                                 |         |
| to latex.tex . . . . .                   | 978     | added . . . . .                                               | 1111    |
| Warnings and errors now directly         |         | @makespecialcolbox: Command                                   |         |
| coded . . . . .                          | 978     | added . . . . .                                               | 1112    |
| 1993-11-23 ltdefns.dtx                   | LaTeX2e | 1993-11-29 lplain.dtx                                         | LaTeX2e |
| @argdef: Macro added . . . . .           | 84      | General: All accents in decimals;                             |         |
| @ifundefined: Redefined to remove a      |         | suggested by Paul Taylor . . . . .                            | 32      |
| trailing \fi . . . . .                   | 106     | 1993-11-30 ltoutput.dtx v1.0c                                 |         |
| @newcommand: Macro added . . . . .       | 84      | \f1@tracemessage: Commands                                    |         |
| @newenv: Macro interface changed .       | 87      | added . . . . .                                               | 1146    |
| @xargdef: Macro interface changed .      | 84      | 1993-12-01 fontdef.dtx                                        | v2.1a   |
| @yargd@f: Avoid \@?@? token . . . . .    | 85      | General: Update for LaTeX2e . . . . .                         | 662     |
| Macro interface changed . . . . .        | 85      | 1993-12-01 ltoutput.dtx v1.0e                                 |         |
|                                          |         | @reinserts: Command added . . . . .                           | 1112    |

|                                                                          |     |                                      |
|--------------------------------------------------------------------------|-----|--------------------------------------|
| 1993-12-03 ltboxes.dtx v0.1a                                             |     | 1993-12-05 ltfloat.dtx LaTeXe        |
| \@argrsbox: macro removed .....                                          | 835 | \@dblfloataplacement: Command        |
| \@begin@tempboxa: macro added ..                                         | 820 | changed ..... 921                    |
| \@end@tempboxa: macro added .....                                        | 820 | \@xfloat: Command changed ..... 915  |
| \@iirsbox: redefined to support<br>\height ..... 835                     |     | 1993-12-05 ltoutput.dtx v1.0f        |
| \@imakebox: macro modified .....                                         | 821 | \@addtobot: Command changed .. 1125  |
| \@iirsbox: redefined to support<br>\height ..... 835                     |     | \@addtocurcol: Command changed 1127  |
| \@isavebox: color support .....                                          | 824 | \@addtobdblcol: Command changed 1137 |
| extra group ..... 824                                                    |     | \@addtonextcol: Command changed 1134 |
| \@isavepicbox: extra group .....                                         | 824 | \@addtotoporbot: Command             |
| \@makebox: default changed from x to<br>c ..... 820                      |     | changed ..... 1126                   |
| \@makepicbox: macro modified .....                                       | 821 | \@boxfpsbit: Command added .. 1149   |
| \@savebox: default c not x .....                                         | 824 | \@flcheckspace: Command added 1151   |
| \bm@b: macros added .....                                                | 820 | \@flsetnum: Command added ... 1150   |
| \endlrbox: macro added .....                                             | 825 | \@flsettextmin: Command added 1151   |
| \fbox: extra group .....                                                 | 825 | \@flstop: Commands added .... 1147   |
| \lrbbox: color support .....                                             | 824 | \@flupdates: Command added .. 1152   |
| macro added ..... 824                                                    |     | \@fpsadddefault: Command added 1148  |
| \makebox: modified .....                                                 | 819 | \@getfpsbit: Command added .. 1149   |
| \mbox: extra group .....                                                 | 820 | \@opcol: Command changed .... 1110   |
| \minipage: Redefined to support extra<br>optional arguments .....        | 831 | Hook added ..... 1110                |
| \newsavebox: Pass the whole of arg 1<br>to \@ifdefinable .....           | 823 | \@outputpage: Command changed 1114   |
| \parbox: Redefined to support extra<br>optional arguments .....          | 828 | \@resethfps: Command added .. 1150   |
| \raisebox: redefined to support<br>\height ..... 834                     |     | \@setfloattypecounts: Command        |
| \sbox: color support .....                                               | 824 | added ..... 1149                     |
| extra group ..... 824                                                    |     | \@setfpsbit: Command added .. 1150   |
| \set@color: color support .....                                          | 822 | \@shipoutsetup: Command added 1114   |
| macro added ..... 822                                                    |     | \@startcolumn: Command changed 1120  |
| 1993-12-03 ltclass.dtx v0.2i                                             |     | \@startdblcolumn: Command            |
| \@cls@pkg: Name changed to avoid<br>clash with output routine. .... 1008 |     | changed ..... 1120                   |
| General: \@onlypreamble: Many<br>commands declared. .... 978             |     | \@testfp: Command added ..... 1149   |
| Removed obsolete<br>\@documentclass .....                                | 978 | \@textfloatsheight: Commands         |
| 1993-12-03 lterror.dtx v1.0b                                             |     | added ..... 1148                     |
| \@latexerr: Set<br>\c@errorcontextlines to -1 .... 394                   |     | \@topnewpage: Commands changed 1102  |
| 1993-12-03 ltfsini.dtx v2.1a                                             |     | \@tryfcolumn: Command changed 1121   |
| General: update for LaTeXe .... 634                                      |     | \@writesetup: \@startpagehook        |
| 1993-12-04 ltfilehook.dtx v0.9b                                          |     | added ..... 1114                     |
| \unqu@tefilef@und: Macro added 1046                                      |     | \output: Command changed .... 1104   |
| 1993-12-04 ltfiles.dtx v0.9b                                             |     | 1993-12-06 ltclass.dtx v0.2k         |
| \@iinput: Macro reimplemented ... 460                                    |     | \ExecuteOptions: Preserve            |
| \@input: Macro reimplemented ... 461                                     |     | \CurrentOption ..... 995             |
| \IfFileExists@: Macro added .... 457                                     |     | 1993-12-06 ltoutput.dtx v1.0f        |
| \input: Macro reimplemented ... 460                                      |     | \@specialoutput: Unboxing of 255     |
|                                                                          |     | added to rescue writes ..... 1104    |
|                                                                          |     | 1993-12-06 ltoutput.dtx v1.0g        |
|                                                                          |     | \@topnewpage: \@floatplacement       |
|                                                                          |     | placement bug fixed ..... 1102       |
|                                                                          |     | 1993-12-07 ltclass.dtx v0.2l         |
|                                                                          |     | \ProvidesFile: Macro added .... 989  |
|                                                                          |     | 1993-12-07 ltclass.dtx v0.2m         |
|                                                                          |     | \load@onefilewithoptions: Reset      |
|                                                                          |     | \CurrentOption ..... 1001            |

|                                                                                                |                                                                                                                                                                              |            |
|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1993-12-07 ltoutenc.dtx 1.1                                                                    | \maybe@ic@: Macro and name<br>changed .....                                                                                                                                  | 691        |
| General: Protected all special<br>characters with \string. ....                                | 469                                                                                                                                                                          |            |
| 1993-12-07 ltoutenc.dtx v1.1                                                                   | \sw@slant: Macro changed .....                                                                                                                                               | 692        |
| General: Made all character numbers<br>decimal. ....                                           | 466                                                                                                                                                                          |            |
| Removed a lot of equal signs and<br>the like. ....                                             | 466                                                                                                                                                                          |            |
| 1993-12-08 ltboxes.dtx v0.1b                                                                   | \textup: Macros changed .....                                                                                                                                                | 689        |
| \@begin@tempboxa: Extra braces for<br>color support (braces removed<br>from other macros) .... | 820                                                                                                                                                                          |            |
| \@irsbox: fix typo .....                                                                       | 835                                                                                                                                                                          |            |
| \@parboxto: \endgraf added due to<br>extra group in \@begin@tempboxa                           | 828                                                                                                                                                                          |            |
| \lrbbox: move \endpefalse out of the<br>inner group ....                                       | 824                                                                                                                                                                          |            |
| 1993-12-08 ltfntcmd.dtx v2.1b                                                                  | \ref: Macro reimplemented .....                                                                                                                                              | 740        |
| General: Macros \rm, \bf and \sf<br>moved to classes.dtx ....                                  | 694                                                                                                                                                                          |            |
| 1993-12-08 ltlists.dtx LaTeX2e                                                                 | \@item: use \sbox to support colour .....                                                                                                                                    | 815        |
| 1993-12-08 ltspace.dtx LaTeX2e                                                                 | \@bsphack: Command reimplemented .....                                                                                                                                       | 428        |
| Command reimplemented; late<br>birthday present for Chris .....                                | 428                                                                                                                                                                          |            |
| \@vbsphack: Command added .....                                                                | 431                                                                                                                                                                          |            |
| 1993-12-09 ltboxes.dtx v0.1c                                                                   | \@irsbox: fix another typo .....                                                                                                                                             | 835        |
| 1993-12-09 ltclass.dtx v0.2n                                                                   | \documentstyle: input 209<br>compatibility file. ....                                                                                                                        | 995        |
| 1993-12-09 ltfiles.dtx v0.9e                                                                   | \document: Hook added .....                                                                                                                                                  | 444        |
| 1993-12-09 ltmiscen.dtx v0.9e                                                                  | \enddocument: Hook added .....                                                                                                                                               | 761        |
| 1993-12-10 ltoutenc.dtx v1.2                                                                   | General: Added source code for<br>t1enc.sty. ....                                                                                                                            | 466        |
| 1993-12-11 ltfntcmd.dtx v3.0a                                                                  | General: Complete reworking of all<br>text commands, using just one<br>creator function .....                                                                                | 686        |
|                                                                                                | italic correction now put in front of<br>penalty before glue .....                                                                                                           | 686        |
|                                                                                                | newcommands replaced by defs ..                                                                                                                                              | 686        |
|                                                                                                | newfontswitch command corrected<br>and changed .....                                                                                                                         | 686        |
|                                                                                                | \DeclareTextFontCommand: Macro<br>changed .....                                                                                                                              | 688        |
|                                                                                                | \emph: Macro changed .....                                                                                                                                                   | 689        |
|                                                                                                | \fix@penalty: Macro added .....                                                                                                                                              | 692        |
|                                                                                                | \maybe@ic: Macro name changed ..                                                                                                                                             | 691        |
|                                                                                                | \textup: Macro and name<br>changed .....                                                                                                                                     | 691        |
|                                                                                                | \sw@slant: Macro changed .....                                                                                                                                               | 692        |
|                                                                                                | \textup: Macros changed .....                                                                                                                                                | 689        |
| 1993-12-11 ltmath.dtx v0.9g                                                                    | General: Added a group around the<br>first argument of \frac to prevent<br>changes (for example font changes)<br>from modifying the contents of the<br>second argument. .... | 795        |
| 1993-12-11 ltoutenc.dtx v1.2a                                                                  | General: Corrected for t1enc, math. ....                                                                                                                                     | 466        |
| 1993-12-11 ltsect.dtx LaTeX2e                                                                  | \author: Added default .....                                                                                                                                                 | 898        |
|                                                                                                | \title: Added default .....                                                                                                                                                  | 898        |
| 1993-12-11 ltxref.dtx LaTeX2e                                                                  | \@setref: Macro added .....                                                                                                                                                  | 740        |
|                                                                                                | \pageref: Macro reimplemented .....                                                                                                                                          | 740        |
|                                                                                                | \ref: Macro reimplemented .....                                                                                                                                              | 740        |
| 1993-12-12 ltoutput.dtx v1.0h                                                                  | \@cf1b: boxmaxdepth setting moved<br>defs changed to lets .....                                                                                                              | 1119       |
|                                                                                                | \@cfilt: name changed .....                                                                                                                                                  | 1119       |
|                                                                                                | \@doclearpage: defs changed to<br>lets .....                                                                                                                                 | 1109, 1110 |
|                                                                                                | \@makecol: defs changed to lets ..                                                                                                                                           | 1111       |
|                                                                                                | \@resethfps: Warnings added:<br>minimal .....                                                                                                                                | 1150       |
|                                                                                                | \@startdblcolumn: defs changed to<br>lets .....                                                                                                                              | 1120, 1121 |
|                                                                                                | \@topnewpage: braces removed ..                                                                                                                                              | 1103       |
|                                                                                                | \@tryfcolumn: defs changed to lets ..                                                                                                                                        | 1122       |
|                                                                                                | \fl@tracemessage: Commands<br>changed .....                                                                                                                                  | 1146       |
| 1993-12-13 ltclass.dtx v0.2o                                                                   | General: Removed setting<br>\errorcontextlines (now in<br>latex.tex) .....                                                                                                   | 978        |
|                                                                                                | \documentstyle: compatibility file<br>now latex209.sty. ....                                                                                                                 | 995        |
|                                                                                                | \usepackage: Fixed error handling ..                                                                                                                                         | 997        |
| 1993-12-13 ltdirchk.dtx v0.2a                                                                  | General: on the 'docstrip' pass, do not<br>check openin path .....                                                                                                           | 10         |
|                                                                                                | \IfFileExists: Removed interactive<br>prompting for current directory<br>syntax .....                                                                                        | 10         |
|                                                                                                | \strip@prefix: modified, name<br>changed from \stripmeaning. ....                                                                                                            | 5          |
| 1993-12-13 ltlists.dtx latex2e                                                                 | \trivlist: Initialised \@itemlabel .....                                                                                                                                     | 810        |
| 1993-12-13 ltmiscen.dtx v0.9h                                                                  | \noligs: Readded \noligs .....                                                                                                                                               | 784        |
|                                                                                                | \verbatim: Readded \noligs ..                                                                                                                                                | 778        |

|                                                                |                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Removed optional argument of<br>\item ..... 778                | Removed all the hackery for use in<br>\DeclareFontEncoding, and redid<br>everything using<br>\DeclareTextFoo. .... 482, 484                                                                                                                                       |
| center: Removed optional argument<br>of \item ..... 775        | Removed the catcode hackery, since<br>the file is only read as a package in<br>the preamble, and removed all the<br>messages on the screen, which just<br>confuse users. Replaced them by<br>the appropriate \ProvidesPackage<br>commands. Added XXXenc. .... 469 |
| flushleft: Removed optional<br>argument of \item ..... 776     |                                                                                                                                                                                                                                                                   |
| flushright: Removed optional<br>argument of \item ..... 777    |                                                                                                                                                                                                                                                                   |
| 1993-12-13 ltoutenc.dtx v1.2b                                  | 1993-12-17 ltoutenc.dtx v1.3                                                                                                                                                                                                                                      |
| General: Corrected file name in driver<br>code. .... 466       | General: Added<br>\EncodingSpecificAccent,<br>\EncodingSpecificAccentedLetter<br>and \EncodingSpecificCommand. 466                                                                                                                                                |
| 1993-12-13 ltab.dtx latex2e                                    | Made Rokicki's encoding a proper<br>encoding scheme rather than a<br>variant of OT1. .... 466                                                                                                                                                                     |
| \tabbing: Removed optional argument<br>of \item ..... 844      | 1993-12-17 ltoutput.dtx v1.0j                                                                                                                                                                                                                                     |
| 1993-12-14 ltoutput.dtx v1.0i                                  | \copcol: Hook removed ..... 1110                                                                                                                                                                                                                                  |
| General: Section added to declare all<br>parameters ..... 1158 | \@specialoutput: Page room test<br>added ..... 1105                                                                                                                                                                                                               |
| 1993-12-15 ltboxes.dtx v0.1d                                   | \@topnewpage: check for vsize too<br>small added ..... 1102                                                                                                                                                                                                       |
| \@iminipage: Changed default from<br>'c' to 's' ..... 831      | Page room test added ..... 1104                                                                                                                                                                                                                                   |
| \@iparbox: Changed default from 'c'<br>to 's' ..... 828        | \@writesetup: —and then removed 1114                                                                                                                                                                                                                              |
| \minipage: Changed default from 'c'<br>to 's' ..... 831        | \f@tracemessage: tracefloatvals<br>made a document command ... 1146                                                                                                                                                                                               |
| extra space removed. ..... 831                                 | 1993-12-17 ltpage.dtx LaTeX2e                                                                                                                                                                                                                                     |
| \parbox: Changed default from 'c' to<br>'s' ..... 828          | \mark: Removed init \mark at begin<br>document, since it doesn't work. 972                                                                                                                                                                                        |
| 1993-12-15 ltclass.dtx v0.2p                                   | \rightmark: Stopgap solution to mark<br>\leftmark and \rightmark work<br>without initializing mark until the<br>problem is solved. .... 972                                                                                                                       |
| General: Removed extra 's' from<br>\@@warnings ..... 978       | 1993-12-18 ltoutenc.dtx 1.3b                                                                                                                                                                                                                                      |
| 1993-12-16 ltlogos.dtx LaTeX2e                                 | General: Fixed typos with<br>\ProvidesPackage lines. Added<br>the \NeedsTeXFormat line. Added<br>the last argument to<br>\DeclareEncoding. Moved the use<br>of the encodings to after their<br>declaration. .... 469                                              |
| \LaTeXe: Extended logo by DPC ... 441                          | Replaced the missing last argument<br>to \DeclareFontEncoding. 482, 484                                                                                                                                                                                           |
| 1993-12-16 ltmath.dtx v0.9i                                    | 1993-12-18 ltoutenc.dtx 1.3c                                                                                                                                                                                                                                      |
| \@eqnctr: use \refstepcounter<br>instead of shortcut ..... 796 | General: Rewrote for the new syntax<br>of \EncodingSpecific. ... 482, 484                                                                                                                                                                                         |
| General: use \refstepcounter instead<br>of shortcut ..... 795  | Split \EncodingSpecificAccent up<br>into \EncodingSpecific and<br>\DeclareAccent. .... 470                                                                                                                                                                        |
| 1993-12-16 ltmiscen.dtx v0.9i                                  | 1993-12-18 ltoutenc.dtx v1.3a                                                                                                                                                                                                                                     |
| General: \literal added ..... 784                              | General: Replaced OT3 by XXX ... 466                                                                                                                                                                                                                              |
| 1993-12-16 ltpage.dtx LaTeX2e                                  |                                                                                                                                                                                                                                                                   |
| \mark: Init \mark at begin document 972                        |                                                                                                                                                                                                                                                                   |
| 1993-12-16 ltspace.dtx LaTeX2e                                 |                                                                                                                                                                                                                                                                   |
| \@bsphack: Corrected optimisation . 428                        |                                                                                                                                                                                                                                                                   |
| 1993-12-16 ltab.dtx latex2e                                    |                                                                                                                                                                                                                                                                   |
| \@xhline: Measure from middle of<br>vertical rules ..... 859   |                                                                                                                                                                                                                                                                   |
| 1993-12-17 ltclass.dtx v0.2q                                   |                                                                                                                                                                                                                                                                   |
| \@documentclasshook: Macro added 978                           |                                                                                                                                                                                                                                                                   |
| \@fileswithoptions: Add<br>\@compatibility hook ..... 999      |                                                                                                                                                                                                                                                                   |
| \documentstyle: Match Alan's new<br>code. ..... 995            |                                                                                                                                                                                                                                                                   |
| 1993-12-17 ltoutenc.dtx 1.3                                    |                                                                                                                                                                                                                                                                   |
| General: Added this section ..... 470                          |                                                                                                                                                                                                                                                                   |

|                                                                                                                                                                   |                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 1993-12-18 ltoutenc.dtx v1.3b                                                                                                                                     | 1994-01-14 ltdirchk.dtx v0.2d                                                  |
| General: Corrected typos. .... 466                                                                                                                                | \IfFileExists: Close the texsys.aux<br>output stream ..... 10                  |
| Replaced the missing last argument<br>to \DeclareFontEncoding. .... 466                                                                                           |                                                                                |
| 1993-12-18 ltoutenc.dtx v1.3c                                                                                                                                     | 1994-01-15 lfiles.dtx v0.9o                                                    |
| General: A new syntax, separating<br>accent-definitions from<br>encoding-specific definitions, and<br>allowing encoding-specific<br>\chardef, \let, etc. .... 466 | \document: move \@preamblecmds<br>after document hook ..... 446                |
| Rewrote for the new syntax of<br>\EncodingSpecific. .... 466                                                                                                      |                                                                                |
| 1993-12-18 ltoutenc.dtx v1.3d                                                                                                                                     | 1994-01-17 ltclass.dtx v0.2s                                                   |
| General: Some T1 stuff had drifted<br>into the OT1 file. .... 466                                                                                                 | \@fileswithoptions: Modify to<br>reduce parameter stack usage ... 999          |
| 1993-12-18 ltpage.dtx LaTeXe                                                                                                                                      | General: Added many more<br>\@onlypreamble commands .... 978                   |
| \sloppy: Added \emergencystretch 972                                                                                                                              | Wrapped long lines to column 72 978                                            |
| 1993-12-19 ltclass.dtx v0.2r                                                                                                                                      | \load@onefile@withoptions: Modify<br>to reduce parameter stack usage 1004      |
| \endfilecontents: Different message<br>when ignoring a file ..... 1010                                                                                            |                                                                                |
| 1993-12-19 lftncmd.dtx v3.0b                                                                                                                                      | 1994-01-17 lfiles.dtx LaTeXe                                                   |
| General: \opdef command added .. 686                                                                                                                              | \listfiles: New Version, adds '.tex'<br>if needed, and lines up columns .. 463 |
| Added by ASAJ. .... 694                                                                                                                                           |                                                                                |
| Made \newfontswitch produce an<br>error if command already exists,<br>and added \renewfontswitch,<br>ASAJ ..... 686                                               | 1994-01-17 ltfssbas.dtx v2.1a                                                  |
| Other tidying ..... 686                                                                                                                                           | General: New math font setup .... 523                                          |
| Some more tidying done ..... 686                                                                                                                                  | \curr@math@size: New math font<br>setup ..... 536                              |
| Untidying added, so this is now a<br>TEMPORARY version. .... 686                                                                                                  | \everydisplay: New math font setup 535                                         |
| Wording changes by CAR. .... 694                                                                                                                                  | \everymath: New math font setup .. 535                                         |
| \DeclareOldFontCommand: Corrected<br>and tidied ..... 693                                                                                                         | \frozen@everydisplay: New math<br>font setup ..... 535                         |
| \DeclareTextFontCommand: Corrected<br>and tidied ..... 688                                                                                                        | \frozen@everymath: New math font<br>setup ..... 535                            |
| 1993-12-19 ltspace.dtx LaTeXe                                                                                                                                     | \math@version: New math font setup 534                                         |
| \bsphack: There seem to be problems<br>with selfmade birthday presents . 429                                                                                      | 1994-01-17 lfsini.dtx v2.1e                                                    |
| 1993-12-20 ltdefs.dtx LaTeXe                                                                                                                                      | \not@math@alphabet: Message<br>changed ..... 655                               |
| \reargdef: Kept old version of<br>\reargdef, for array.sty ..... 85                                                                                               | 1994-01-17 lfsstrc.dtx v2.3a                                                   |
| 1993-12-20 lfiles.dtx v0.9m                                                                                                                                       | General: New math font setup .... 572                                          |
| \obsoletefile: Added this<br>command, removed<br>@oldfilewarning ..... 463                                                                                        | \check@mathfonts: New math font<br>setup ..... 582                             |
| 1994-01-05 fontdef.dtx v2.1d                                                                                                                                      | \glb@currsize: New math font setup 580                                         |
| General: Removed nf prefix from file<br>names. .... 664                                                                                                           | \restglb@settings: New math font<br>setup ..... 583                            |
| 1994-01-13 ltmath.dtx v0.9o                                                                                                                                       | 1994-01-18 ltbibl.dtx LaTeXe                                                   |
| \eqnacr: correcting 0.9i ..... 796                                                                                                                                | \bibliography: Use \input@ so<br>include files are listed. .... 939            |
| General: correcting 0.9i ..... 795                                                                                                                                |                                                                                |
|                                                                                                                                                                   | 1994-01-18 ltclass.dtx v0.2t                                                   |
|                                                                                                                                                                   | \ifclassloaded: Fix typo<br>\pkgetension ..... 983                             |
|                                                                                                                                                                   | 1994-01-18 lfilehook.dtx v0.9p                                                 |
|                                                                                                                                                                   | \unqu@tefilef@und: New Definition 1046                                         |
|                                                                                                                                                                   | 1994-01-18 lfiles.dtx v0.9p                                                    |
|                                                                                                                                                                   | \iffileonpath: Macro added .... 459                                            |
|                                                                                                                                                                   | \input: do not use a different<br>definition for \input@path .... 461          |
|                                                                                                                                                                   | \input@: Macro added ..... 461                                                 |
|                                                                                                                                                                   | \IfFileExists@: New Definition ... 457                                         |
|                                                                                                                                                                   | \includeonly: Use \input@ so<br>include files are listed. .... 450             |

|                                                                             |  |                                                                                                         |
|-----------------------------------------------------------------------------|--|---------------------------------------------------------------------------------------------------------|
| 1994-01-18 ltfssini.dtx v2.1f                                               |  | 1994-01-25 ltfssbas.dtx v2.1b                                                                           |
| \@not@math@alphabet: Message<br>corrected ..... 655                         |  | \math@version: Corrections for math<br>setup ..... 534                                                  |
| 1994-01-18 ltmiscen.dtx v0.9p                                               |  | 1994-01-25 ltmath.dtx LaTeXe                                                                            |
| \@verbatim: Add<br>\global\@inlabelfalse ..... 778                          |  | \bordermatrix: Removed \p@renwd. 790                                                                    |
| Only add \penalty if in hmode .. 778                                        |  |                                                                                                         |
| 1994-01-19 fontdef.dtx v2.1e                                                |  | 1994-01-26 lfsstrc.dtx v2.3c                                                                            |
| General: Added missing setting for<br>symbols in bold version. ..... 669    |  | \check@mathfonts: Correct trace info<br>placement ..... 582                                             |
| 1994-01-19 ltdirchk.dtx v0.2e                                               |  | \restglb@settings: Correct trace info<br>placement ..... 583                                            |
| \IfFileExists: name changed from<br>\test ..... 9                           |  | 1994-01-27 lfntcmd.dtx v3.1a                                                                            |
| \input@path: No longer check that an<br>empty group is in the path ..... 10 |  | \nocorrlist: Only ., used as default<br>for cm fonts ..... 693                                          |
| \strip@prefix: name changed from<br>\strip@meaning, to match NFSS. 5        |  |                                                                                                         |
| 1994-01-19 ltmath.dtx v1.0n classes                                         |  | 1994-01-29 ltclass.dtx v0.2v                                                                            |
| \mathindent: Deferred setting of<br>\mathindent ..... 798                   |  | \@unprocessedoptions: Macro<br>added ..... 1009                                                         |
| 1994-01-20 ltdirchk.dtx v0.2f                                               |  | \load@onefile@withoptions: All<br>options raise error if no<br>\ProcessOptions appears ..... 1005       |
| General: \copytexsys and the<br>texsys.new file removed ..... 8             |  | 1994-01-31 ltdefns.dtx v0.2w                                                                            |
| Modify all of ltxcheck ..... 13                                             |  | \g@addto@macro: Use toks register to<br>avoid ‘hash’ problems ..... 112                                 |
| \IfFileExists: \copytexsys<br>removed ..... 10                              |  | 1994-01-31 ltfiles.dtx v0.9t                                                                            |
| 1994-01-21 ltclass.dtx v0.2u                                                |  | \document: set \normalsize or<br>\normalsize if necessary ..... 445                                     |
| \documentstyle: compatibility file<br>now latex209.def. ..... 995           |  | 1994-01-31 lfntcmd.dtx v3.1b                                                                            |
| 1994-01-21 ltdirchk.dtx v0.2g                                               |  | General: \normalsize no longer<br>defined ..... 686                                                     |
| General: Improve documentation,<br>reorganize docstrip module ..... 1       |  | 1994-02-01 ltpage.dtx LaTeXe                                                                            |
| \filename@parse: Minor changes, and<br>add Mac version () ..... 11          |  | \pagestyle: (DPC) Modify to get<br>nicer error message ..... 969                                        |
| \today: Name changed from \stamp,<br>to save memory ..... 9                 |  | \thispagestyle: (DPC) Modify to get<br>nicer error message ..... 970                                    |
| 1994-01-21 ltfloat.dtx LaTeXe                                               |  | 1994-02-02 ltclass.dtx v0.2x                                                                            |
| \@xfloat: Added missing percent<br>characters. ..... 915                    |  | \load@onefile@withoptions: Only<br>run the hook and options check if<br>the file was loaded. ..... 1005 |
| 1994-01-21 ltmiscen.dtx v0.9s                                               |  | 1994-02-03 ltoutput.dtx v1.0k                                                                           |
| \verbatim@font: Removed<br>unnecessary category code<br>hackery. ..... 779  |  | \makespecialcolbox: correct<br>mistakes in the documentation. 1113                                      |
| 1994-01-24 ltdirchk.dtx v0.2h                                               |  | 1994-02-07 ltclass.dtx v0.2y                                                                            |
| \IfFileExists: Stop testing once<br>texsys.aux has been found ..... 9       |  | \files@withoptions: Run<br>\compatibility on the first class<br>to start (not the first to finish). 999 |
| 1994-01-24 ltpage.dtx LaTeXe                                                |  | \@ifclasswith: Add extra ,s so ‘two’<br>is not matched with ‘twocolumn’ 985                             |
| \pagestyle: (DPC) Complain if<br>pagestyle is undefined. ..... 969          |  | \ProcessOptions*: Add extra ,s so<br>‘two’ is not matched with<br>‘twocolumn’ ..... 992                 |
| 1994-01-25 ltdirchk.dtx v0.2i                                               |  | 1994-02-07 ltfssbas.dtx v2.1c                                                                           |
| General: Protect against looping on<br>\@input and \@end. ..... 3           |  | \DeclareFontEncoding: revert catcode<br>settings earlier ..... 527                                      |
|                                                                             |  | \DeclareFontShape@: revert catcode<br>settings earlier ..... 524                                        |

|            |                                                                                                                                                                                 |      |            |                                                                |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|----------------------------------------------------------------|
| 1994-02-08 | ltoutput.dtx v1.0k                                                                                                                                                              |      | 1994-03-04 | ltab.dtx v1.0a                                                 |
|            | \@makespecialcolbox: boxmaxdepth<br>setting added . . . . .                                                                                                                     | 1113 |            | General: Initial version, split from<br>latex.dtx . . . . .    |
|            | boxmaxdepth setting removed .                                                                                                                                                   | 1112 |            | 837                                                            |
|            | General: Documentation and tasks<br>tidied. . . . .                                                                                                                             | 1087 | 1994-03-04 | ltvers.dtx v1.0a                                               |
| 1994-02-10 | ltclass.dtx v0.2z                                                                                                                                                               |      |            | General: Initial version, split from<br>latex.dtx . . . . .    |
|            | \@documentclasshook: Changed the<br>name from \@compatibility to<br>\@documentclasshook, and added<br>the check for whether<br>\@normalsize has been defined.<br>ASA.J. . . . . | 978  |            | 37                                                             |
|            | \@fileswithoptions: Renamed<br>\@compatibility to<br>\@documentclasshook. ASA.J. . .                                                                                            | 999  | 1994-03-07 | ltboxes.dtx v0.1a                                              |
| 1994-02-10 | ltfssbas.dtx v2.1d                                                                                                                                                              |      |            | \@mpfootnotetext: Extra group for<br>color . . . . .           |
|            | \addto@hook: Made \addto@hook<br>long. . . . .                                                                                                                                  | 549  |            | 832                                                            |
| 1994-02-10 | ltfscmp.dtx v2.1d                                                                                                                                                               |      | 1994-03-07 | ltboxes.dtx v1.0a                                              |
|            | \scan@fontshape: scan away stuff<br>after pt . . . . .                                                                                                                          | 598  |            | General: Unify format with other<br>Kernel files . . . . .     |
| 1994-02-22 | ltfssini.dtx v2.1g                                                                                                                                                              |      | 1994-03-07 | ltdefns.dtx v1.0a                                              |
|            | General: Correct error message . . .                                                                                                                                            | 659  |            | \@@italiccorr: Macro added . . . . .                           |
| 1994-02-24 | ltfssbas.dtx v2.1e                                                                                                                                                              |      | 1994-03-07 | ltfiles.dtx v1.0a                                              |
|            | \DeclareFontShape: Separate<br>restoration of catcodes for fd<br>cmds . . . . .                                                                                                 | 524  |            | General: Initial version, split from<br>latex.dtx . . . . .    |
|            | \define@newfont: Separate<br>restoration of catcodes for fd<br>cmds . . . . .                                                                                                   | 537  |            | 442                                                            |
|            | \nfss@catcodes: Separate restoration<br>of catcodes for fd cmds . . . . .                                                                                                       | 538  |            | Long lines wrapped to 72 columns . . . . .                     |
| 1994-02-25 | ltdirchk.dtx v0.2j                                                                                                                                                              |      | 1994-03-07 | ltfinal.dtx v0.1a                                              |
|            | General: Remove need for drv file . . .                                                                                                                                         | 1    |            | General: Add code from the old<br>dump.dtx . . . . .           |
| 1994-03-01 | ltdirchk.dtx v0.2k                                                                                                                                                              |      |            | 1190                                                           |
|            | General: Add unstripped module, so<br>that dircheck.dtx may be used<br>with initex . . . . .                                                                                    | 1    |            | Initial version, split from latex.dtx . . . . .                |
| 1994-03-02 | ltboxes.dtx v0.1e                                                                                                                                                               |      |            | 1174                                                           |
|            | General: Add 2ekernel module . . . . .                                                                                                                                          | 819  |            | move code here from lhyphen.dtx . . . . .                      |
|            | Remove need for drv file . . . . .                                                                                                                                              | 819  |            | 1180                                                           |
| 1994-03-02 | ltclass.dtx v0.3a                                                                                                                                                               |      |            | Remove oldcomments<br>environment . . . . .                    |
|            | General: Remove need for driver file . . .                                                                                                                                      | 978  |            | 1174                                                           |
| 1994-03-03 | ltboxes.dtx v0.1f                                                                                                                                                               |      |            | use \InputIfFileExists not<br>\IfFileExists . . . . .          |
|            | \@irsbox: Replaced a missing \else . . .                                                                                                                                        | 835  |            | 1180                                                           |
| 1994-03-04 | ltfloat.dtx v1.0a                                                                                                                                                               |      | 1994-03-07 | ltfloat.dtx v1.0a                                              |
|            | General: Initial version, split from<br>latex.dtx . . . . .                                                                                                                     | 911  |            | \@endfloatbox: (DPC) Extra group<br>for colour . . . . .       |
| 1994-03-04 | ltsect.dtx v1.0a                                                                                                                                                                |      |            | 920                                                            |
|            | General: Initial version, split from<br>latex.dtx . . . . .                                                                                                                     | 898  |            | \@footnotetext: (DPC) Extra group<br>for colour . . . . .      |
|            |                                                                                                                                                                                 |      |            | 930                                                            |
|            |                                                                                                                                                                                 |      |            | \@xfloat: (DPC) Extra group for<br>colour . . . . .            |
|            |                                                                                                                                                                                 |      |            | 916                                                            |
| 1994-03-07 | lhyphen.dtx v0.1c                                                                                                                                                               |      | 1994-03-07 | lthyphen.dtx v0.1c                                             |
|            | General: move the 2ekernel code to<br>ltfinal.dtx . . . . .                                                                                                                     | 1172 |            | General: move the 2ekernel code to<br>ltfinal.dtx . . . . .    |
| 1994-03-07 | ltlength.dtx v1.0a                                                                                                                                                              |      |            | 1172                                                           |
|            | \@settodim: (DPC) Extra group for<br>colour . . . . .                                                                                                                           | 521  | 1994-03-07 | ltlength.dtx v1.0a                                             |
| 1994-03-07 | ltlists.dtx v1.0a                                                                                                                                                               |      |            | \@settodim: (DPC) Extra group for<br>colour . . . . .          |
|            | General: Initial version, split from<br>latex.dtx . . . . .                                                                                                                     | 802  |            | 802                                                            |
|            | Long lines wrapped to 72 columns . . . . .                                                                                                                                      | 802  |            | Long lines wrapped to 72 columns . . . . .                     |
| 1994-03-07 | ltpage.dtx v1.0a                                                                                                                                                                |      | 1994-03-07 | ltpage.dtx v1.0a                                               |
|            | General: Initial version, split from<br>ltherest.dtx . . . . .                                                                                                                  | 969  |            | General: Initial version, split from<br>ltherest.dtx . . . . . |
| 1994-03-07 | lt pictur.dtx v0.1a                                                                                                                                                             |      |            | 969                                                            |
|            | General: Initial version, split from<br>latex.dtx . . . . .                                                                                                                     | 862  |            | General: Initial version, split from<br>latex.dtx . . . . .    |
|            | Long lines wrapped to 72 columns . . . . .                                                                                                                                      | 862  |            | 862                                                            |
| 1994-03-07 | ltsect.dtx v1.0a                                                                                                                                                                |      |            | \@hangfrom: (DPC) Extra groups for<br>colour . . . . .         |
|            | \@hangfrom: (DPC) Extra groups for<br>colour . . . . .                                                                                                                          | 905  |            | 905                                                            |

|                                                                                                                               |     |                                                                           |            |
|-------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------------|------------|
| 1994-03-07 lttab.dtx v1.0a                                                                                                    |     | 1994-03-13 ltfilehook.dtx v0.3b                                           |            |
| General: Long lines wrapped to 72<br>columns . . . . .                                                                        | 837 | \unqu@tefilef@und: Use new cmd<br>\@addtofilelist . . . . .               | 1046       |
| 1994-03-08 ltclass.dtx v0.3b                                                                                                  |     | 1994-03-13 ltfiles.dtx LaTeX2e                                            |            |
| General: Modify driver code into ‘new<br>style’ . . . . .                                                                     | 978 | \@addtofilelist: Macro added . . . . .                                    | 463        |
| 1994-03-08 ltdirchk.dtx v1.0a                                                                                                 |     | \listfiles: Reset \@addtofilelist<br>at begin document . . . . .          | 464        |
| General: Reorganize driver module<br>into ‘new style’ . . . . .                                                               | 1   | 1994-03-13 ltfssbas.dtx v2.1g                                             |            |
| 1994-03-08 ltplain.dtx v1.0a                                                                                                  |     | General: add 2ekernel module to omit<br>repeated code . . . . .           | 523        |
| General: Remove need for a driver file.                                                                                       | 14  | 1994-03-13 ltfssdcl.dtx v2.1c                                             |            |
| 1994-03-10 ltfssbas.dtx v2.2f                                                                                                 |     | General: add 2ekernel module to omit<br>repeated code . . . . .           | 602        |
| \math@egroup: Changed<br>\begingroup/\endgroup to<br>\bgrou@egroup. . . . .                                                   | 546 | 1994-03-14 ltboxes.dtx v1.0b                                              |            |
| 1994-03-11 ltfssdcl.dtx v2.1b                                                                                                 |     | \@isavebox: Use \color@setgroup . . . . .                                 | 824        |
| \DeclareSymbolFontAlphabet@:<br>Added check against use of<br>alphabet switch outside of math<br>mode. . . . .                | 632 | \@isavepicbox: Use<br>\color@setgroup . . . . .                           | 824        |
| \SetMathAlphabet@: Changed<br>parameter template in temporary<br>macro to catch check add below.                              | 621 | \color@begingroup: macro added for<br>color support . . . . .             | 822        |
| 1994-03-12 ltclass.dtx v0.3c                                                                                                  |     | \color@endgroup: macro added for<br>color support . . . . .               | 823        |
| General: Change name from docclass<br>to ltclass . . . . .                                                                    | 978 | \lrbox: Use \color@setgroup . . . . .                                     | 824        |
| \ProvidesFile: Add \wlog . . . . .                                                                                            | 989 | \sbox: Use \color@setgroup . . . . .                                      | 824        |
| \ProvidesPackage: Add \wlog . . . . .                                                                                         | 987 | 1994-03-14 ltfloat.dtx 1.0c                                               |            |
| use \gttempa . . . . .                                                                                                        | 987 | \@xympar: (DPC) Use<br>\color@begingroup . . . . .                        | 925        |
| 1994-03-12 ltdefns.dtx v1.0b                                                                                                  |     | 1994-03-14 ltfloat.dtx v1.0c                                              |            |
| \@reargdef: New defn, in terms of<br>\@yargdef . . . . .                                                                      | 85  | \@endfloatbox: (DPC) Use<br>\color@endgroup . . . . .                     | 920        |
| \@yargd@f: Name changed from<br>\XXX@argdef . . . . .                                                                         | 85  | \@footnotetext: (DPC) Use<br>\color@begingroup, add<br>\endgraf . . . . . | 930        |
| 1994-03-12 ltdirchk.dtx v1.0b                                                                                                 |     | \@savemarbox: (DPC) Use<br>\color@begingroup . . . . .                    | 924        |
| General: Change name from<br>dircheck.dtx . . . . .                                                                           | 1   | \@xfloat: (DPC) Use<br>\color@begingroup . . . . .                        | 916        |
| Minor edits to the typeouts in<br>ltxcheck . . . . .                                                                          | 1   | 1994-03-15 ltfiles.dtx LaTeX2e                                            |            |
| 1994-03-12 ltfloat.dtx v1.0b                                                                                                  |     | \@missingfileerror: Quit on x or X<br>just like a real error . . . . .    | 461        |
| \@savemarbox: (DPC) Extra group for<br>colour . . . . .                                                                       | 924 | 1994-03-15 lftntcmd.dtx v3.2a                                             |            |
| \@xympar: (DPC) Extra bgroup for<br>colour . . . . .                                                                          | 925 | General: Adapted to mass formatting . . . . .                             | 686        |
| 1994-03-12 ltplain.dtx v1.0b                                                                                                  |     | Changed \v to \@italiccorr . . . . .                                      | 686        |
| General: Name changed from lplain.<br>The end of an era . . . . .                                                             | 14  | Removed \crenewfontswitch . . . . .                                       | 686        |
| 1994-03-12 ltplain.dtx v1.0e                                                                                                  |     | Removed defs of short-forms and all<br>sizes except \normalize . . . . .  | 686        |
| General: Replaced remaining width,<br>height, depth by L <sup>A</sup> T <sub>E</sub> X macro<br>names to save tokens. . . . . | 14  | 1994-03-15 ltoutput.dtx v1.0l                                             |            |
| 1994-03-13 ltcntrl.dtx v1.0c                                                                                                  |     | \@addtocr: Changed \addvspace<br>to \vskip . . . . .                      | 1129, 1132 |
| \@tfor: (DPC) Add \@tf@r so a single<br>group is correctly treated. . . . .                                                   | 387 | \@combinedblfloats: Removed<br>boxmaxdepth setting. . . . .               | 1120       |
|                                                                                                                               |     | \@makecol: \maxdepth changed to<br>\@maxdepth . . . . .                   | 1111       |
|                                                                                                                               |     | Removed boxmaxdepth setting. . . . .                                      | 1112       |

|                                                                                                       |      |                                                                                                                              |      |
|-------------------------------------------------------------------------------------------------------|------|------------------------------------------------------------------------------------------------------------------------------|------|
| \@makespecialcolbox: Removed<br>boxmaxdepth setting. . . . .                                          | 1113 | 1994-03-28 ltab.dtx v1.0b<br>General: Improve documentation . . . . .                                                        | 837  |
| \@topnewpage: Corrected and<br>amended warning message . . . . .                                      | 1103 | 1994-03-28 ltthm.dtx v1.0a<br>General: Initial version, split from<br>latex.dtx . . . . .                                    | 894  |
| Warning added: it should be<br>improved . . . . .                                                     | 1104 | 1994-03-29 ltcnts.dtx v1.0c<br>General: Create file from parts of<br>ltmisen and ltherest. . . . .                           | 512  |
| General: Added some warnings when<br>page gets full of top floats. . . . .                            | 1087 | 1994-03-29 ltlenth.dtx v1.0c<br>General: Create file ltcntlen from parts<br>of ltmisen and ltherest. . . . .                 | 521  |
| Driver added and further tidying. . . . .                                                             | 1087 | 1994-03-29 ltmisen.dtx v1.0d<br>General: Remove counter macros to<br>ltcntlen . . . . .                                      | 760  |
| Removed duplicated code and<br>corrected docstrip options. . . . .                                    | 1087 | 1994-03-29 ltpageno.dtx v1.0c<br>General: Create file ltcntlen from parts<br>of ltmisen and ltherest. . . . .                | 737  |
| Some boxmaxdepth settings<br>removed. . . . .                                                         | 1087 | 1994-03-29 ltxref.dtx v1.0c<br>General: Create file ltcntlen from parts<br>of ltmisen and ltherest. . . . .                  | 738  |
| 1994-03-16 ltclass.dtx v0.3f<br>General: Add pkgindoc package . . . . .                               | 1026 | 1994-03-31 ltbibl.dtx v1.0a<br>General: Initial version of ltidxbib.dtx,<br>split from ltherest.dtx . . . . .                | 937  |
| 1994-03-16 ltfiles.dtx LaTeXe<br>\listfiles: Move this code directly<br>into \document . . . . .      | 464  | 1994-03-31 ltidglo.dtx v1.0a<br>General: Initial version of ltidxbib.dtx,<br>split from ltherest.dtx . . . . .               | 934  |
| 1994-03-16 ltfiles.dtx v1.0c<br>\document: (DPC) directly add file list<br>settings . . . . .         | 446  | 1994-04-09 ltcnts.dtx v1.0d<br>\newctr: \@nocnterr now has<br>counter name argument . . . . .                                | 513  |
| 1994-03-16 ltmisen.dtx v1.0b<br>\verbatim: Remove<br>\global\@inlabelfalse again. . . . .             | 778  | \addtocounter: \@nocnterr now has<br>counter name argument . . . . .                                                         | 513  |
| 1994-03-28 ltalloc.dtx v1.0d<br>General: Redefinition of 'new'<br>allocations removed. . . . .        | 382  | \setcounter: \@nocnterr now has<br>counter name argument . . . . .                                                           | 513  |
| 1994-03-28 ltdirchk.dtx v1.0d<br>General: Improve documentation . . . . .                             | 1    | \stepcounter: Use \addtocounter to<br>have name checked . . . . .                                                            | 514  |
| 1994-03-28 lterror.dtx v1.0d<br>\@invalidchar: (DPC) Comment out<br>(use catcode15 instead) . . . . . | 398  | 1994-04-09 ltthm.dtx v1.0b<br>\othm: Use standard counter error<br>message (FMi) . . . . .                                   | 896  |
| General: Remove test for<br>\inputlineno undefined. . . . .                                           | 394  | 1994-04-11 ltclass.dtx v0.3g<br>\endfilecontents: Add star form,<br>don't write \endinput at the end<br>of the file. . . . . | 1010 |
| 1994-03-28 ltfiles.dtx v1.0d<br>\document: (DPC) Use \normalsize<br>not \@normalsize . . . . .        | 445  | \ProvidesFile: Protect against weird<br>catcodes. . . . .                                                                    | 989  |
| (DPC) remove<br>\@normalsize check . . . . .                                                          | 445  | 1994-04-11 ltfssbas.dtx v2.1h<br>General: Added \defaultscriptratio<br>and \defaultscriptscriptratio.<br>ASAJ. . . . .       | 523  |
| 1994-03-28 ltfloat.dtx v1.0b<br>\caption: Use \normalsize not<br>\@normalsize . . . . .               | 914  | \defaultscriptratio: Macro added . . . . .                                                                                   | 547  |
| General: Split further from<br>ltherest.dtx . . . . .                                                 | 911  | \defaultscriptscriptratio: Macro<br>added . . . . .                                                                          | 547  |
| 1994-03-28 ltlists.dtx v1.0b<br>General: Improve documentation . . . . .                              | 801  | 1994-04-12 ltboxes.dtx v1.0c<br>General: Remove \@acci, now defined<br>in ltplain.dtx . . . . .                              | 829  |
| 1994-03-28 ltmisen.dtx v1.0c<br>General: Improve Documentation . . . . .                              | 760  |                                                                                                                              |      |
| 1994-03-28 ltplain.dtx v1.0c<br>\newlanguage: Remove some \outer<br>declarations. . . . .             | 17   |                                                                                                                              |      |
| 1994-03-28 ltsect.dtx v1.0b<br>General: Split further from<br>ltherest.dtx . . . . .                  | 898  |                                                                                                                              |      |

|                                                                                                                              |      |                                                                                                                             |      |
|------------------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------|------|
| Remove \@dischyp, now defined in<br>ltinit.dtx . . . . .                                                                     | 829  | 1994-04-18 ltfssdcl.dtx v2.1d<br>General: Removed surplus<br>\no@alphabet@error (see<br>fam.dtx) . . . . .                  | 602  |
| 1994-04-12 ltdefns.dtx v1.0g<br>\@dischyp: Define \@dischyp, was<br>previously in ltboxes.dtx . . . . .                      | 110  | 1994-04-18 lfsstrc.dtx v2.3d<br>General: Changed to new<br>error/warning scheme . . . . .                                   | 572  |
| 1994-04-12 lplain.dtx v1.0d<br>General: Define \@acci . . . . .                                                              | 32   | \font@submax: Changed dimen to<br>macro . . . . .                                                                           | 591  |
| 1994-04-12 ltvers.dtx v1.0b<br>General: Have version info generated<br>automatically. . . . .                                | 37   | \fontsubfuzz: Changed dimen to<br>macro . . . . .                                                                           | 591  |
| 1994-04-14 lfntcmd.dtx v3.2b<br>General: Macros renamed to<br>non-private forms, JB . . . . .                                | 686  | \subst@size: \font@submax and<br>\fontsubfuzz now macros . . . . .                                                          | 592  |
| \DeclareOldFontCommand: Renamed<br>from \@newfontswitch . . . . .                                                            | 693  | 1994-04-19 ltpage.dtx v1.0b<br>General: Improve documentation . . .                                                         | 969  |
| 1994-04-15 ltboxes.dtx v1.0d<br>\@isavebox: Added missing percent<br>character. . . . .                                      | 824  | 1994-04-20 lfntcmd.dtx v3.3a<br>General: Documentation up-dated . .                                                         | 686  |
| 1994-04-17 ltcnts.dtx v1.0e<br>\@newctr: Use \@nocnterr instead<br>of \@nocntr . . . . .                                     | 513  | New implementation of \nocorr . . .                                                                                         | 686  |
| \addtocounter: Use \@nocnterr<br>instead of \@nocntr . . . . .                                                               | 513  | \check@nocorr@: Macros added . . .                                                                                          | 690  |
| \setcounter: Use \@nocnterr<br>instead of \@nocntr . . . . .                                                                 | 513  | \maybe@ic@: \nocorr etc removed<br>from list of tokens to check, leaving<br>only punctuation characters . . . . .           | 691  |
| 1994-04-17 lterror.dtx v1.0h<br>\@nocnterr: New name for error<br>message, old error message<br>(without arg) kept . . . . . | 395  | 1994-04-20 ltmiscen.dtx v1.0e<br>\enddocument@kernel@warnings:<br>Changed logic for producing<br>warning messages . . . . . | 763  |
| 1994-04-17 ltthm.dtx v1.0c<br>\@othm: Use new std counter error<br>message (FMi) . . . . .                                   | 896  | 1994-04-21 ltboxes.dtx v1.0e<br>\@iiminipage: Extra \bgroup for<br>color . . . . .                                          | 831  |
| \@othm: Use new std counter error<br>message (FMi) . . . . .                                                                 | 896  | \@mpfootnotetext: Extra \endgraf<br>for color . . . . .                                                                     | 832  |
| 1994-04-18 ltfinal.dtx v0.1b<br>General: Initialise \textheight,<br>\textwidth and page style . . . . .                      | 1177 | \endminipage: Extra \egroup for<br>color . . . . .                                                                          | 832  |
| 1994-04-18 ltfloat.dtx v1.0d<br>\@footnotetext: (DPC) Remove<br>Colour support . . . . .                                     | 930  | 1994-04-21 ltfinal.dtx v0.1c<br>General: Added comments, set the<br>catcodes of 128–255. . . . .                            | 1174 |
| \@savemarbox: (DPC) Remove Colour<br>support . . . . .                                                                       | 924  | 1994-04-22 ltfssini.dtx v2.1g<br>\not@math@alphabet: Message<br>changed again . . . . .                                     | 655  |
| 1994-04-18 ltfssbas.dtx v2.1i<br>General: Macro \no@alphabet@help<br>removed again . . . . .                                 | 523  | 1994-04-23 ltfinal.dtx v0.1d<br>General: Check that \font@submax is<br>still zero . . . . .                                 | 1174 |
| \calculate@math@sizes: Changed<br>message to log only . . . . .                                                              | 546  | 1994-04-24 ltoutput.dtx v1.0m<br>\resetfps: Number 2 changed to<br>\tw@ . . . . .                                           | 1150 |
| \no@alphabet@error: Use std LaTeX<br>error macro . . . . .                                                                   | 523  | Warning changed . . . . .                                                                                                   | 1150 |
| 1994-04-18 ltfssdcl.dtx ???<br>\DeclareMathAlphabet: Pass correct<br>arg (2 not 3) . . . . .                                 | 619  | \@specialoutput: Message changed to<br>give more info and ‘top’ removed .                                                   | 1105 |
|                                                                                                                              |      | \@topnewpage: Message changed to<br>give more info . . . . .                                                                | 1104 |
|                                                                                                                              |      | Warning message removed as it will<br>be generated later . . . . .                                                          | 1103 |
|                                                                                                                              |      | General: Changed \@normalsize to<br>\normalsize. . . . .                                                                    | 1087 |

|                                                                                                                                                                   |            |                                                                              |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------|-----|
| Corrected unverbed commands in documentation. . . . .                                                                                                             | 1087       | Removed Rokicki's OT1 variant encoding. Moved the driver to the top. . . . . | 469 |
| Removed some long lines and other aesthetic changes. . . . .                                                                                                      | 1087       |                                                                              |     |
| Warning messages changed/corrected. . . . .                                                                                                                       | 1087       |                                                                              |     |
| 1994-04-24 ltpictur.dtx v0.1b                                                                                                                                     |            |                                                                              |     |
| General: Removed surplus spaces after <code>\hbox to</code> in several cases . . . . .                                                                            | 862        |                                                                              |     |
| 1994-04-25 ltclass.dtx v0.3h                                                                                                                                      |            |                                                                              |     |
| General: Removed spurious extra 's at the end of error messages . . . . .                                                                                         | 978        |                                                                              |     |
| 1994-04-25 ltfloat.dtx v1.0e                                                                                                                                      |            |                                                                              |     |
| <code>\@largefloatcheck</code> : Changed warning message to give more info . . . . .                                                                              | 920        |                                                                              |     |
| Command added . . . . .                                                                                                                                           | 920        |                                                                              |     |
| General: Changed warning messages . . . . .                                                                                                                       | 911        |                                                                              |     |
| Removed obsolete tracing code . . . . .                                                                                                                           | 911        |                                                                              |     |
| 1994-04-27 lfsstrc.dtx v2.3e                                                                                                                                      |            |                                                                              |     |
| General: Corrected item that was forgotten in last change. . . . .                                                                                                | 572        |                                                                              |     |
| 1994-04-28 lterror.dtx v1.0j                                                                                                                                      |            |                                                                              |     |
| <code>\@inmatherr</code> : Macro added . . . . .                                                                                                                  | 398        |                                                                              |     |
| 1994-04-28 lterror.dtx v1.1c                                                                                                                                      |            |                                                                              |     |
| <code>\@inmatherr</code> : Replaced <code>\noexpand</code> with <code>\protect</code> . . . . .                                                                   | 398        |                                                                              |     |
| 1994-04-28 ltfssdcl.dtx v2.1e                                                                                                                                     |            |                                                                              |     |
| General: Removed all <code>\uppercase</code> in hex num parsing macros . . . . .                                                                                  | 602        |                                                                              |     |
| 1994-04-28 ltlists.dtx v1.0c                                                                                                                                      |            |                                                                              |     |
| <code>\item</code> : Replaced <code>\@ltxnomath</code> by <code>\@inmatherr</code> . . . . .                                                                      | 813        |                                                                              |     |
| 1994-04-28 ltpictur.dtx v0.1c                                                                                                                                     |            |                                                                              |     |
| <code>\@multiput</code> : (DPC) Macro added . . . . .                                                                                                             | 866        |                                                                              |     |
| General: bezier curves added . . . . .                                                                                                                            | 889        |                                                                              |     |
| <code>\@multiput</code> : (DPC) Ignore spaces between )(. . . . .                                                                                                 | 865        |                                                                              |     |
| <code>\picture</code> : (DPC) Ignore spaces before ( . . . . .                                                                                                    | 864        |                                                                              |     |
| 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) . . . . .                                                                                                   | 978        |                                                                              |     |
| 1994-04-29 ltmiscen.dtx v1.0f                                                                                                                                     |            |                                                                              |     |
| <code>\@verbatim</code> : <code>\leavevmode</code> added . . . . .                                                                                                | 778        |                                                                              |     |
| Change to <code>\everypar</code> added . . . . .                                                                                                                  | 778        |                                                                              |     |
| 1994-04-29 ltoutenc.dtx 1.4a                                                                                                                                      |            |                                                                              |     |
| General: Removed <code>\EncodingSpecific</code> . Renamed all the commands. Added <code>\DeclareTextGlyph</code> and <code>\UndeclareTextCommand</code> . . . . . | 470        |                                                                              |     |
| 1994-04-30 ltfntcmd.dtx v3.3b                                                                                                                                     |            |                                                                              |     |
| General: Documentation up-dated and tidied . . . . .                                                                                                              | 686        |                                                                              |     |
| Prefix <code>frag@</code> changed to <code>frag</code> in <code>\@protecteddef</code> . . . . .                                                                   | 686        |                                                                              |     |
| Title changed . . . . .                                                                                                                                           | 686        |                                                                              |     |
| Warning changed to info message in <code>\@protecteddef</code> . . . . .                                                                                          | 686        |                                                                              |     |
| 1994-04-30 ltoutput.dtx v1.0n                                                                                                                                     |            |                                                                              |     |
| <code>\@activechar@info</code> :                                                                                                                                  |            |                                                                              |     |
| <code>\@activechar@warning</code> changed to <code>\@activechar@info</code> . . . . .                                                                             | 1114       |                                                                              |     |
| <code>\@combinedblfloats</code> : Removed rule in <code>topnewpage</code> case . . . . .                                                                          | 1120       |                                                                              |     |
| <code>\@emptycol</code> : Empty column action added: <code>\@emptycol</code> . . . . .                                                                            | 1102       |                                                                              |     |
| <code>\@fisetnum</code> : Rogue space removed . . . . .                                                                                                           | 1151       |                                                                              |     |
| <code>\@specialoutput</code> : Cut-off point changed to <code>2\baselineskip</code> . . . . .                                                                     | 1105       |                                                                              |     |
| Empty column action added: <code>\@emptycol</code> . . . . .                                                                                                      | 1105       |                                                                              |     |
| Extra empty column added for twocolumn case . . . . .                                                                                                             | 1105       |                                                                              |     |
| Extra empty column added for twocolumn case (wrong, see below) . . . . .                                                                                          | 1105       |                                                                              |     |
| <code>\@topnewpage</code> : Added setting of <code>\col@number</code> . . . . .                                                                                   | 1102, 1103 |                                                                              |     |
| Cut-off point changed to <code>3\baselineskip</code> . . . . .                                                                                                    | 1104       |                                                                              |     |
| Empty column action added: <code>\@emptycol</code> . . . . .                                                                                                      | 1104       |                                                                              |     |
| Message changed for Frank . . . . .                                                                                                                               | 1104       |                                                                              |     |
| General: <code>\@activechar@warning</code> changed to an info message. . . . .                                                                                    | 1087       |                                                                              |     |
| Added <code>\col@number</code> . . . . .                                                                                                                          | 1087       |                                                                              |     |
| Documentation tidied. . . . .                                                                                                                                     | 1087       |                                                                              |     |
| Empty column action added. . . . .                                                                                                                                | 1087       |                                                                              |     |
| Fixed bug from <code>\dblfigrule</code> with <code>\@topnewpage</code> . . . . .                                                                                  | 1087       |                                                                              |     |
| Full of floats action improved. . . . .                                                                                                                           | 1087       |                                                                              |     |
| <code>\col@number</code> : Added <code>\col@number</code> . . . . .                                                                                               | 1099       |                                                                              |     |
| <code>\onecolumn</code> : Added setting of <code>\col@number</code> . . . . .                                                                                     | 1101       |                                                                              |     |
| 1994-05-01 lterror.dtx v1.0k                                                                                                                                      |            |                                                                              |     |
| <code>\@latexerr</code> : (CAR) Added draft <code>\@latexinfo</code> . . . . .                                                                                    | 394        |                                                                              |     |
| 1994-05-01 ltoutenc.dtx 1.4a                                                                                                                                      |            |                                                                              |     |
| General: Added the <code>\a</code> command. . . . .                                                                                                               | 478        |                                                                              |     |
| Added the <code>\SaveAtCatcode</code> and <code>\RestoreAtCatcode</code> commands. . . . .                                                                        | 482        |                                                                              |     |

|                                                                                                                        |          |                                                                                 |      |
|------------------------------------------------------------------------------------------------------------------------|----------|---------------------------------------------------------------------------------|------|
| Removed the uc/lc table settings,<br>since the T1 uc/lc table is now the<br>default. . . . .                           | 490      | Removed unnecessary braces from<br>arguments of \@ifnextchar . . . . .          | 911  |
| Rewrote for the new syntax. . . . .                                                                                    | 482, 484 | \end@dblfloat: \largefloatcheck<br>added . . . . .                              | 919  |
| 1994-05-01 ltoutenc.dtx v1.4a                                                                                          |          | \end@float: (CAR) Added<br>\largefloatcheck . . . . .                           | 918  |
| General: Removed Rokicki's<br>encoding. . . . .                                                                        | 466      | 1994-05-03 ltfssdcl.dtx v2.1f                                                   |      |
| Renamed the commands, removed<br>the \EncodingSpecific command.<br>Turned all slots into decimal.<br>Added \a. . . . . | 466      | General: Renamed<br>\@DeclarMathDelimiter to<br>\@DeclarMathDelimiter . . . . . | 602  |
| 1994-05-02 ltcntrl.dtx v1.0l                                                                                           |          | \@item: \hskip changed to \kern . . . . .                                       | 814  |
| \break@tfor: Macro added (from<br>ltfiles.dtx) . . . . .                                                               | 387      | \item: Removed superfluous braces . . . . .                                     | 813  |
| 1994-05-02 ltdefns.dtx v1.1f                                                                                           |          | 1994-05-03 ltmiscen.dtx v1.0h                                                   |      |
| \renewcommand: Removed surplus<br>\space in error . . . . .                                                            | 86       | \centercr: \badcrerr replaced by<br>\nolnerr . . . . .                          | 774  |
| \renewenvironment: Removed surplus<br>\space in error . . . . .                                                        | 87       | 1994-05-03 ltab.dtx v1.0d                                                       |      |
| 1994-05-02 ltfiles.dtx v1.0f                                                                                           |          | \endpbox: Use \finalstrut based<br>on depth of \arstrutbox . . . . .            | 860  |
| \iffileonpath: \break@loop<br>renamed to \break@tfor . . . . .                                                         | 459      | 1994-05-04 ltclass.dtx v1.0b                                                    |      |
| \obsoletefile: Make<br>\onlypreamble . . . . .                                                                         | 463      | \NeedsTeXFormat: Changed wording of<br>the warning . . . . .                    | 998  |
| 1994-05-02 ltfinal.dtx v0.1e                                                                                           |          | 1994-05-04 lterror.dtx v1.0m                                                    |      |
| General: Added setting the 'letter'<br>catcodes. . . . .                                                               | 1186     | \badcrerr: Error message removed . . . . .                                      | 397  |
| Added setting the 'other'<br>catcodes. . . . .                                                                         | 1186     | 1994-05-05 ltbibl.dtx v1.0c                                                     |      |
| Added setting the special<br>catcodes. . . . .                                                                         | 1186     | \citex: Set switch for warning and<br>end of run. . . . .                       | 939  |
| Made slot 127 illegal . . . . .                                                                                        | 1186     | \nocite: Do not write page number in<br>\nocite warning message. . . . .        | 940  |
| Set all the catcodes . . . . .                                                                                         | 1174     | Set switch for warning and end of<br>run. . . . .                               | 940  |
| 1994-05-02 ltfinal.dtx v0.1f                                                                                           |          | 1994-05-05 ltfinal.dtx v0.1g                                                    |      |
| General: Set the catcode of<br>control-J. . . . .                                                                      | 1186     | General: Added empty errhelp. . . . .                                           | 1174 |
| 1994-05-02 ltmiscen.dtx v1.0g                                                                                          |          | \errhelp: Set error help empty. . . . .                                         | 1191 |
| General: Changed 91 to 1991 and<br>moved some bits . . . . .                                                           | 760      | 1994-05-05 lfntcmd.dtx v3.3c                                                    |      |
| 1994-05-02 ltoutput.dtx v1.0o                                                                                          |          | \@math@egroup: Corrected<br>\fontswitch and added saved<br>versions . . . . .   | 693  |
| \resethfps: Code shortened . . . . .                                                                                   | 1150     | General: Corrected \fontswitch . . . . .                                        | 686  |
| General: Code of \resethfps<br>shortened. . . . .                                                                      | 1087     | 1994-05-05 ltmiscen.dtx v1.0i                                                   |      |
| 1994-05-03 ltbibl.dtx v1.0b                                                                                            |          | General: Removed braces from<br>ifnextchar and ifstar arguments . . . . .       | 760  |
| \nocite: Make \nocite issue a<br>warning for an undefined citation<br>key. . . . .                                     | 940      | 1994-05-07 ltab.dtx v1.0c                                                       |      |
| 1994-05-03 ltfinal.dtx v0.1f                                                                                           |          | \maxtab: Changed \firsttab to<br>\chardef . . . . .                             | 841  |
| General: Set the catcode of control-J<br>to be 'other', for use in messages.                                           | 1174     | Changed \maxtab to \chardef . . . . .                                           | 841  |
| 1994-05-03 ltfloat.dtx v1.0f                                                                                           |          | General: Removed definition of \+ . . . . .                                     | 837  |
| General: (CAR) Added<br>\largefloatcheck . . . . .                                                                     | 911      | Removed surplus braces from<br>\ifnextchar constructs . . . . .                 | 837  |
| 1994-05-08 lfntcmd.dtx v3.3d                                                                                           |          | 1994-05-08 lfntcmd.dtx v3.3d                                                    |      |
| General: Removed<br>\undefinedfonterror . . . . .                                                                      | 686      | General: Removed<br>\undefinedfonterror . . . . .                               | 694  |
| \normalsize: Removed<br>\undefinedfonterror . . . . .                                                                  | 694      |                                                                                 |      |

|            |                                                                                                                                                                                                     |                                                       |     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|-----|
| 1994-05-09 | ltfntcmd.dtx v3.3f                                                                                                                                                                                  | Renamed the ‘package’ module to<br>‘teststy’. . . . . | 469 |
|            | General: Replaced all <code>\next</code> by<br><code>\@let@token</code> and undo change<br>3.3e, whatever that was. . . . .                                                                         | 686                                                   |     |
| 1994-05-10 | ltdefns.dtx v1.0n                                                                                                                                                                                   |                                                       |     |
|            | General: (ASAJ) Added<br><code>\DeclareProtectedCommand</code> . . . . .                                                                                                                            | 80                                                    |     |
|            | Added <code>\DeclareProtectedCommand</code>                                                                                                                                                         | 90                                                    |     |
|            | Removed braces around<br><code>\@ifundefined</code> argument. ASAJ. . . . .                                                                                                                         | 86                                                    |     |
|            | <code>\makeatother</code> : Added <code>\makeatletter</code><br>and <code>\makeatother</code> ASAJ. . . . .                                                                                         | 109                                                   |     |
| 1994-05-10 | lterror.dtx v1.0n                                                                                                                                                                                   |                                                       |     |
|            | <code>\@latexerr</code> : (ASAJ) Added extra<br>blank lines to <code>\@latexerr</code> . . . . .                                                                                                    | 394                                                   |     |
| 1994-05-10 | ltmiscen.dtx v1.0j                                                                                                                                                                                  |                                                       |     |
|            | <code>\@sverb</code> : Slight change in error<br>message text. . . . .                                                                                                                              | 781                                                   |     |
| 1994-05-11 | ltboxes.dtx v1.0f                                                                                                                                                                                   |                                                       |     |
|            | <code>\begin@tempboxa</code> : Use new<br><code>\color@setgroup</code> concept. . . . .                                                                                                             | 820                                                   |     |
|            | <code>\@iimminipage</code> : Use new<br><code>\color@setgroup</code> concept. . . . .                                                                                                               | 831                                                   |     |
|            | <code>\@mpfootnotetext</code> : Use new<br><code>\color@setgroup</code> concept. . . . .                                                                                                            | 832                                                   |     |
|            | Use new <code>\normalcolor</code> and<br><code>\finalstrut</code> . . . . .                                                                                                                         | 832                                                   |     |
|            | General: Superfluous braces removed<br>from several commands . . . . .                                                                                                                              | 819                                                   |     |
|            | <code>\color@setgroup</code> : macro added for<br>color support . . . . .                                                                                                                           | 822                                                   |     |
|            | <code>\endminipage</code> : Use new<br><code>\color@setgroup</code> concept. . . . .                                                                                                                | 832                                                   |     |
| 1994-05-11 | ltclass.dtx v1.0c                                                                                                                                                                                   |                                                       |     |
|            | <code>\endfilecontents</code> : Add checks for<br>form feed and tab . . . . .                                                                                                                       | 1010                                                  |     |
| 1994-05-11 | ltdirchk.dtx v1.0e                                                                                                                                                                                  |                                                       |     |
|            | General: Add <code>\ProvidesFile</code> as used<br>in fd files. . . . .                                                                                                                             | 4                                                     |     |
| 1994-05-11 | lterror.dtx v1.0o                                                                                                                                                                                   |                                                       |     |
|            | <code>\@latexerr</code> : (ASAJ) Removed one of<br>the extra blank lines to<br><code>\@latexerr</code> . . . . .                                                                                    | 394                                                   |     |
| 1994-05-11 | ltlogos.dtx v1.0o                                                                                                                                                                                   |                                                       |     |
|            | <code>\LaTeX</code> : Use<br><code>\DeclareProtectedCommand</code> .<br>ASAJ. . . . .                                                                                                               | 441                                                   |     |
|            | <code>\LaTeXe</code> : Use<br><code>\DeclareProtectedCommand</code> .<br>ASAJ. . . . .                                                                                                              | 441                                                   |     |
| 1994-05-11 | ltoutenc.dtx 1.5a                                                                                                                                                                                   |                                                       |     |
|            | General: Made T1 and OT1 generate<br>packages rather than def files.                                                                                                                                |                                                       |     |
| 1994-05-11 | ltoutenc.dtx v1.5a                                                                                                                                                                                  |                                                       |     |
|            | General: Reimplemented<br><code>\DeclareTextCommand</code> using<br><code>\@changed@cmd</code> and<br><code>\DeclareProtectedCommand</code> . . . . .                                               | 470                                                   |     |
|            | Renamed the commands again.<br>Made the encoding part of the<br>command syntax. Added the<br><code>\DeclareTextCommand</code> interface.<br>Used<br><code>\DeclareProtectedCommand</code> . . . . . | 466                                                   |     |
|            | <code>\DeclareTextAccent</code> : Reimplemented<br>using <code>\DeclareTextCommand</code> . . . . .                                                                                                 | 472                                                   |     |
| 1994-05-11 | ltspace.dtx v1.0o                                                                                                                                                                                   |                                                       |     |
|            | <code>\hspace</code> : Use<br><code>\DeclareRobustCommand</code> . ASAJ. . . . .                                                                                                                    | 439                                                   |     |
| 1994-05-12 | ltboxes.dtx v1.0g                                                                                                                                                                                   |                                                       |     |
|            | <code>\finalstrut</code> : macro added . . . . .                                                                                                                                                    | 835                                                   |     |
|            | <code>\fbox</code> : New definition, merged with<br><code>\framebox</code> . . . . .                                                                                                                | 825                                                   |     |
|            | <code>\framebox</code> : Merged <code>\fbox</code> and<br><code>\framebox</code> . . . . .                                                                                                          | 826                                                   |     |
|            | <code>\normalcolor</code> : macro added for color<br>support . . . . .                                                                                                                              | 822                                                   |     |
| 1994-05-12 | ltdefns.dtx v1.0p                                                                                                                                                                                   |                                                       |     |
|            | General: (ASAJ) Fixed a bug with<br><code>\relax</code> which was using <code>\gobble</code><br>before defining it. . . . .                                                                         | 80                                                    |     |
|            | Fixed a bug with <code>\relax</code> which<br>was using <code>\gobble</code> before defining<br>it. . . . .                                                                                         | 90                                                    |     |
| 1994-05-12 | ltfssbas.dtx v2.1j                                                                                                                                                                                  |                                                       |     |
|            | General: New baselinestretch concept                                                                                                                                                                | 523                                                   |     |
|            | Replaced hand-protected commands<br>by <code>\DeclareRobustCommand</code> defs                                                                                                                      | 523                                                   |     |
|            | <code>\f@linespread</code> : New macro . . . . .                                                                                                                                                    | 533                                                   |     |
|            | <code>\fontencoding</code> : Use<br><code>\DeclareRobustCommand</code> . . . . .                                                                                                                    | 531                                                   |     |
|            | <code>\fontfamily</code> : Use<br><code>\DeclareRobustCommand</code> . . . . .                                                                                                                      | 532                                                   |     |
|            | <code>\fontseries</code> : Use<br><code>\DeclareRobustCommand</code> . . . . .                                                                                                                      | 532                                                   |     |
|            | <code>\fontshape</code> : Use<br><code>\DeclareRobustCommand</code> . . . . .                                                                                                                       | 532                                                   |     |
|            | <code>\fontsize</code> : Redefined to use<br><code>\set@fontsize</code> . . . . .                                                                                                                   | 533                                                   |     |
|            | <code>\linespread</code> : New macro . . . . .                                                                                                                                                      | 533                                                   |     |
|            | <code>\mathversion</code> : Use<br><code>\DeclareRobustCommand</code> . . . . .                                                                                                                     | 534                                                   |     |
| 1994-05-12 | ltfssdcl.dtx v2.1g                                                                                                                                                                                  |                                                       |     |
|            | General: Allow <code>\relax</code> as undefined<br>command . . . . .                                                                                                                                | 602                                                   |     |

|            |                                                                            |          |                                                                               |      |
|------------|----------------------------------------------------------------------------|----------|-------------------------------------------------------------------------------|------|
|            | Allow \relax'ed cmds to be declared .....                                  | 602      | Coded more efficiently, thanks to FMI. ....                                   | 86   |
| 1994-05-12 | ltfssini.dtx v2.1i                                                         |          | 1994-05-13 ltfssini.dtx v2.1i                                                 |      |
|            | General: Moved \fontencoding to fam.dtx .....                              | 634      | \listfiles: Stop \listfiles being run twice .....                             | 463  |
|            | Moved \fontfamily to fam.dtx .....                                         | 634      | 1994-05-13 ltfssini.dtx v1.0g                                                 |      |
|            | Moved \fontseries to fam.dtx .....                                         | 634      | \document: Added execution of \every@size .....                               | 445  |
|            | Moved \fontshape to fam.dtx .....                                          | 634      | 1994-05-13 ltfinal.dtx v0.1h                                                  |      |
|            | Moved \fontsize to fam.dtx .....                                           | 634      | General: Added package ot1enc, and defined \@acci, \@accii and \@acciii ..... | 1174 |
|            | Moved \mathversion to fam.dtx .....                                        | 634      | 1994-05-13 ltfinal.dtx v1.0h                                                  |      |
|            | Moved \selectfont to tracefntr.dtx .....                                   | 634      | General: Added output enc stuff .....                                         | 1190 |
| 1994-05-12 | ltfsstrc.dtx v2.3f                                                         |          | 1994-05-13 ltfloat.dtx v1.0g                                                  |      |
|            | \selectfont: Use \DeclareRobustCommand .....                               | 576      | \@footnotetext: (DPC) Add new style colour support: \normalcolor .....        | 930  |
| 1994-05-12 | ltoutenc.dtx 1.5a                                                          |          | (DPC) Use \finalstrut .....                                                   | 930  |
|            | General: Removed the \SaveAtCatcode and \RestoreAtCatcode commands .....   | 482      | \@xfloat: (DPC) Use \normalcolor .....                                        | 916  |
|            | Rewrote for the new syntax. ....                                           | 482, 484 | 1994-05-13 lftntcmd.dtx v3.3g                                                 |      |
| 1994-05-12 | ltoutput.dtx v1.0p                                                         |          | General: Replaced \protecteddef by \DeclareRobustCommand .....                | 686  |
|            | \@writesetup: \normalcoloradded .....                                      | 1114     | 1994-05-13 ltfsbas.dtx v2.1k                                                  |      |
|            | General: \normalcoloradded in various places (DPC). ....                   | 1087     | General: Remove File identification 'typeout' .....                           | 523  |
| 1994-05-13 | ltboxes.dtx v1.0h                                                          |          | 1994-05-13 ltfsbas.dtx v2.1l                                                  |      |
|            | \@arrayparboxrestore: New accent system, use \let not \def .....           | 830      | \DeclareFontEncoding: Init encoding change command .....                      | 527  |
| 1994-05-13 | ltcounts.dtx v1.0f                                                         |          | \define@newfont: Use \input@ for fd files .....                               | 537  |
|            | General: Removed \@Ialph .....                                             | 518      | 1994-05-13 ltfsdcl.dtx v2.1h                                                  |      |
|            | Removed \@ialph .....                                                      | 518      | General: Removed file identification typeout .....                            | 602  |
| 1994-05-13 | ltdefns.dtx v1.0q                                                          |          | 1994-05-13 ltfsdcl.dtx v2.1j                                                  |      |
|            | General: (ASAJ) Renamed \DeclareProtectedCommand to \DeclareRobustCommand. |          | General: Removed file identification typeout .....                            | 634  |
|            | Removed \@if@short@command .....                                           | 80       | 1994-05-13 ltfsstrc.dtx v2.3g                                                 |      |
|            | (ASAJ) Replaces \space by ' ' in \csname. ....                             | 80       | General: Removed typeouts as \ProvidesPackage writes to log. ....             | 572  |
|            | Renamed \DeclareProtectedCommand to \DeclareRobustCommand.                 |          | 1994-05-13 ltoutenc.dtx v1.5b                                                 |      |
|            | Removed \@if@short@command .....                                           | 80       | General: Added \{, \} and \\$. ....                                           | 466  |
|            | Moved to after the definition of \@gobble. ....                            | 90       | Renamed \DeclareProtectedCommand to \DeclareRobustCommand .....               | 466  |
| 1994-05-13 | ltdefns.dtx v1.0r                                                          |          | Replaces \space by ' ' in \csname. ....                                       | 466  |
|            | General: (ASAJ) Added logging message to \DeclareProtectedCommand .....    | 80       | 1994-05-13 ltpictur.dtx v0.1d                                                 |      |
|            | Added logging message to \DeclareProtectedCommand .....                    | 90       | General: Removed surplus braces from \@if.. constructions .....               | 862  |
| 1994-05-13 | ltdefns.dtx v1.0s                                                          |          | 1994-05-13 ltab.dtx v1.0d                                                     |      |
|            | General: (ASAJ) Added \@backslashchar. ....                                | 80       | \@contfield: Colour support .....                                             | 843  |
|            | (ASAJ) Coded \@ifdefinable more efficiently. ....                          | 80       | \@startfield: Colour support .....                                            | 843  |
|            |                                                                            |          | \@stopfield: Colour support .....                                             | 843  |
|            |                                                                            |          | \a: moved to ltoutenc .....                                                   | 841  |

|                                        |          |                                        |      |
|----------------------------------------|----------|----------------------------------------|------|
| 1994-05-14 fontdef.dtx v2.1f           |          | (ASAJ) Split from ltinit.dtx.          | 388  |
| General: Removed .def files.           | 664      |                                        |      |
| 1994-05-14 ltfssbas.dtx v2.1m          |          | General: moved output enc stuff to     |      |
| \enc@update: Macro added               | 532      | lffonts                                | 1190 |
| 1994-05-14 ltfssbas.dtx v2.1n          |          | 1994-05-16 ltfssbas.dtx v2.1p          |      |
| General: Set defaults for all \f@...   | 533      | \fontsize: Pass \baselinestretch not   |      |
| \DeclareErrorFont: Don't set           |          | \f@linespread                          | 533  |
| \f@encoding                            | 541      | \linespread: Remove surplus braces     | 533  |
| \DeclareFontEncoding: Log if           |          | 1994-05-16 ltfsini.dtx v2.1m           |      |
| encoding is redeclared                 | 527      | \@acci: Define saved versions of       |      |
| Only init enc change cmd when new      |          | accents                                | 660  |
| encoding                               | 527      | 1994-05-16 ltlogos.dtx v1.1a           |      |
| 1994-05-14 ltfsini.dtx v2.1k           |          | General: (ASAJ) Split from ltinit.dtx. | 441  |
| General: Init error font just before   |          | 1994-05-16 ltmath.dtx v1.0k            |      |
| checking for fontdef.cfg               | 659      | \ensuremath: Use                       |      |
| \reset@font: Remove surplus braces     | 657      | \ DeclareRobustCommand and add         |      |
| 1994-05-14 lftsstrc.dtx v2.3h          |          | extra braces in math mode              | 797  |
| \selectfont: Added \enc@update         | 578      | 1994-05-16 ltoutenc.dtx 1.5h           |      |
| 1994-05-14 ltoutenc.dtx 1.5d           |          | General: \pounds was still using u     |      |
| General: Moved the driver to the top.  | 469      | rather than ui shape.                  | 482  |
| 1994-05-14 ltoutenc.dtx v1.5c          |          | 1994-05-16 ltoutenc.dtx v1.5f          |      |
| General: Added the fontenc package     | 509      | General: enc files now have uc         |      |
| Added the fontenc package.             | 466      | encoding name parts (FMi)              | 466  |
| Fixed a bug which caused an            |          | Revert code so that the encoding       |      |
| infinite loop if \f@encoding was       |          | given is used in                       |      |
| incorrectly set.                       | 466, 470 | \DeclareTextCommand (FMi)              | 466  |
| Moved fontsmp to its own dtx file.     | 466      | 1994-05-16 ltoutenc.dtx v1.5g          |      |
| 1994-05-14 ltoutenc.dtx v1.5d          |          | General: Made fontenc.sty use the new  |      |
| General: Rewrote                       |          | mixed-case encoding files.             | 466  |
| \DeclareTextCommand to define its      |          | Removed the lowercasing of the         |      |
| argument to use the current            |          | filename.                              | 509  |
| encoding by default, rather than       |          | 1994-05-16 ltoutenc.dtx v1.5h          |      |
| the encoding provided to               |          | General: Added \NG, \ng, \TH, \th,     |      |
| \DeclareTextCommand.                   | 466, 470 | \DH, \dh, \DJ and \dj.                 | 466  |
| Tidied up the documentation.           | 466      | Added \r (ring accent) and \k          |      |
| 1994-05-14 ltoutenc.dtx v1.5e          |          | (ogonek) accents.                      | 466  |
| General: Replaced \ENC@cmd by          |          | Fixed a bug with \pounds.              | 466  |
| \ENC-cmd.                              | 466      | Removed \P from the OT1                |      |
| 1994-05-15 ltfssbas.dtx v2.1o          |          | definitions file.                      | 466  |
| General: encoding cmd changed to       |          | 1994-05-16 ltoutenc.dtx v1.5i          |      |
| enc-cmd                                | 523      | General: Fixed a bug with \d.          | 466  |
| 1994-05-16 fontdef.dtx v2.1g           |          | 1994-05-16 ltoutput.dtx v1.0q          |      |
| General: Removed                       |          | \@writesetup: Changed setting of       |      |
| \DeclareFontEncoding for ot1 and       |          | accents (FMi): with the new            |      |
| t1 and input .def files instead        | 664      | encoding setup they can use \let.      |      |
| 1994-05-16 lalloc.dtx v1.1a            |          | It could also use the new internal     |      |
| General: (ASAJ) Split from ltinit.dtx. | 382      | commands?                              | 1115 |
| 1994-05-16 lcntrl.dtx v1.0a            |          | General: Changed setting of accents    |      |
| General: (ASAJ) Split from ltinit.dtx. | 384      | (FMi).                                 | 1087 |
| 1994-05-16 ltdefns.dtx v1.1a           |          | 1994-05-16 ltpar.dtx v1.1a             |      |
| General: (ASAJ) Split from ltinit.dtx. | 80       | General: (ASAJ) Split from ltinit.dtx. | 399  |
| 1994-05-16 lterror.dtx v1.1a           |          | 1994-05-16 ltplain.dtx v1.0h           |      |
| General: (ASAJ) Completely new         |          | General: Comment out encoding          |      |
| error interface.                       | 388      | specific commands                      | 31   |

|                                                                                                                                 |      |                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------|
| Remove \@acci and friends again . . . . .                                                                                       | 32   | 1994-05-18 ltoutenc.dtx v1.5k                                                   |
| Remove unnecessary def for \item . . . . .                                                                                      | 31   | General: Made dotted-i produce ‘i’. . . . . 466                                 |
| \loop: Use Kabelschacht method . . . . .                                                                                        | 29   | Removed braces from \pounds and \dollar. . . . . 466                            |
| \m@th: Remove unnecessary space . . . . .                                                                                       | 31   | Replaced \defaultencoding with \encodingdefault. . . . . 466                    |
| 1994-05-16 ltspace.dtx v1.1a                                                                                                    |      |                                                                                 |
| General: (ASAJ) Split from ltinit.dtx. . . . .                                                                                  | 421  |                                                                                 |
| 1994-05-17 ltclass.dtx v1.0e                                                                                                    |      |                                                                                 |
| \@use@option: Execute option after removing from list, not before . . . . .                                                     | 994  | 1994-05-19 ltbibl.dtx v1.1a                                                     |
| 1994-05-17 ltdefns.dtx 1.1b                                                                                                     |      | General: Initial version of ltbibl.dtx, split from ltidxbib.dtx . . . . . 937   |
| General: (ASAJ) Added the \@protect@... commands. . . . .                                                                       | 90   | 1994-05-19 ltcntlen.dtx v1.1a                                                   |
| 1994-05-17 ltdefns.dtx v1.1b                                                                                                    |      | General: Extracted file from ltcntlen. . . . . 512                              |
| General: (ASAJ) Added definitions for protect. . . . .                                                                          | 80   | 1994-05-19 ltdefns.dtx v1.1d                                                    |
| (ASAJ) Removed warnings and logging to lterror.dtx. . . . .                                                                     | 80   | General: (RmS) Added definitions for \@namedef and \@nameuse again. . . . . 80  |
| Added the discussion of protected commands, defined the values that \protect should have. . . . .                               | 90   | 1994-05-19 ltfinal.dtx v0.1k                                                    |
| 1994-05-17 ltdefns.dtx v1.1c                                                                                                    |      | General: Removed \makeat... . . . . . 1174                                      |
| General: (ASAJ) Redid definitions for protect. . . . .                                                                          | 80   | 1994-05-19 ltidxglo.dtx v1.1a                                                   |
| 1994-05-17 lterror.dtx v1.1b                                                                                                    |      | General: Initial version of ltidxglo.dtx, split from ltidxbib.dtx . . . . . 934 |
| General: (ASAJ) Moved error stuff from ltdefns.dtx. . . . .                                                                     | 388  | 1994-05-19 ltlength.dtx v1.1a                                                   |
| 1994-05-17 ltfssini.dtx v2.1n                                                                                                   |      | General: Extract file ltlength from ltcntlen. . . . . 521                       |
| \copyright: Really add extra braces . . . . .                                                                                   | 657  | 1994-05-19 ltpageno.dtx v1.1a                                                   |
| \@fss@text: Added braces to allow use in subscripts . . . . .                                                                   | 657  | General: Extract file ltpageno from ltcntlen. . . . . 737                       |
| 1994-05-17 ltmath.dtx v1.0i                                                                                                     |      | 1994-05-19 lplain.dtx v0.1k ltfinal                                             |
| General: Replaced \let by \gdef, for indirect definition. . . . .                                                               | 792  | \showoutput: used \maxdimen not 99999 . . . . . 33                              |
| 1994-05-17 ltoutenc.dtx v1.5j                                                                                                   |      | \showoverfull: used \one not 1 . . . . . 33                                     |
| General: Added braces to \pounds so it works as a subscript. . . . .                                                            | 466  | 1994-05-19 ltxref.dtx v1.1a                                                     |
| 1994-05-18 ltdefns.dtx 1.1c                                                                                                     |      | General: Extract file ltxref from ltcntlen. . . . . 738                         |
| General: (ASAJ) Renamed the commands, and removed one which is no longer needed. . . . .                                        | 90   | 1994-05-20 ltdefns.dtx v1.1e                                                    |
| 1994-05-18 ltdefns.dtx v1.1c                                                                                                    |      | General: Changed command name from \@checkcommand to \CheckCommand. . . . . 80  |
| General: Redid the discussion and definitions, in line with the proposed new setting of \protect in the output routine. . . . . | 90   | \CheckCommand: Changed name from \@checkcommand to \CheckCommand. . . . . 88    |
| 1994-05-18 ltfinal.dtx v0.1j                                                                                                    |      | 1994-05-20 lterror.dtx v1.1c                                                    |
| General: Corrected the lccode for d-bar. . . . .                                                                                | 1174 | General: (ASAJ) Added \@latex@info@no@line. . . . . 388                         |
| 1994-05-18 ltlogos.dtx v1.1b                                                                                                    |      | (ASAJ) Added missing full stops. . . . . 388                                    |
| General: (ASAJ) Added the T <sub>E</sub> X logo. . . . .                                                                        | 441  | (ASAJ) Fixed a bug with \@inmatherr. . . . . 388                                |
| (ASAJ) Made the L <sub>T</sub> E <sub>X</sub> 2 <sub>ε</sub> logo use the text font ‘2’ rather than the math font ‘2’. . . . .  | 441  | 1994-05-20 ltfinal.dtx v0.1l                                                    |
|                                                                                                                                 |      | General: Use new font warning commands . . . . . 1181                           |
|                                                                                                                                 |      | 1994-05-20 ltfloat.dtx v1.0h                                                    |
|                                                                                                                                 |      | \@endfloatbox: Restore outer value of @nobreak switch. . . . . 920              |
|                                                                                                                                 |      | \outer@nobreak: Macro added: default is to do nothing. . . . . 920              |

|                                               |      |                                             |      |
|-----------------------------------------------|------|---------------------------------------------|------|
| 1994-05-20 lfntcmd.dtx v3.3h                  |      | 1994-05-22 lterror.dtx v1.2a                |      |
| General: Use new error commands . . . . .     | 686  | General: (ASAJ) Made                        |      |
| 1994-05-20 ltfssbas.dtx v2.1q                 |      | \GenericError, \GenericWarning              |      |
| General: Use new error commands . . . . .     | 523  | and \GenericInfo robust. . . . .            | 388  |
| 1994-05-20 lfsstrc.dtx v2.3i                  |      | (ASAJ) Replaced \\ and tilde by             |      |
| General: Use new error command                |      | \MessageBreak and \space. . . . .           | 388  |
| names . . . . .                               | 572  | (ASAJ) Replaced                             |      |
| 1994-05-20 ltmiscen.dtx v1.0l                 |      | \@generic@message and                       |      |
| \@writefile: Added correct setting of         |      | \@generic@error by                          |      |
| \protect. . . . .                             | 767  | \GenericError, \GenericWarning              |      |
| 1994-05-20 ltmiscen.dtx v1.0m                 |      | and \GenericInfo. . . . .                   | 388  |
| General: Use new warning commands . . . . .   | 760  | (ASAJ) Replaces \string by                  |      |
| 1994-05-20 ltoutput.dtx v1.0s                 |      | \protect in some messages. . . . .          | 388  |
| \@writesetup: Added setting of                |      | 1994-05-22 lterror.dtx v1.2d                |      |
| \protect during \shipout. . . . .             | 1114 | \GenericError: (DPC) Alternative            |      |
| General: Added setting of \protect            |      | version added for old TeXs . . . . .        | 389  |
| during \shipout. . . . .                      | 1087 | (DPC) New version using long                |      |
| 1994-05-20 ltpage.dtx v1.0d                   |      | command name. . . . .                       | 389  |
| \markright: Changed setting for               |      | 1994-05-22 lffloat.dtx v1.0i                |      |
| \protect. . . . .                             | 970  | General: Use new warning commands . . . . . | 911  |
| 1994-05-20 ltsect.dtx v1.0c                   |      | 1994-05-22 ltoutput.dtx v1.0t               |      |
| General: Correct setting of \protect. . . . . | 908  | General: Changed warnings and infos         |      |
| \addcontentsline: Correct setting of          |      | to new commands. . . . .                    | 1087 |
| \protect. . . . .                             | 907  | 1994-05-22 ltpictur.dtx v0.1e               |      |
| 1994-05-21 ltbibl.dtx v1.1b                   |      | General: Use new warning cmd . . . . .      | 862  |
| General: Use new warning commands . . . . .   | 937  | 1994-05-23 ltclass.dtx v1.0h                |      |
| 1994-05-21 lterror.dtx v1.1d                  |      | \NeedsTeXFormat: Don't stop                 |      |
| General: (ASAJ) Made the error                |      | completely when format is wrong . . . . .   | 998  |
| commands robust. . . . .                      | 388  | \usepackage: Remove argument if             |      |
| 1994-05-21 ltfiles.dtx v1.0h                  |      | possible . . . . .                          | 997  |
| General: Use new error commands . . . . .     | 442  | 1994-05-23 ltdirchk.dtx v1.0f               |      |
| 1994-05-21 ltlists.dtx v1.0f                  |      | General: Document \CTeXversion . . . . .    | 1    |
| General: Use new error commands . . . . .     | 801  | 1994-05-23 lfsstrc.dtx v2.3j                |      |
| 1994-05-21 ltmiscen.dtx v1.0n                 |      | General: Removed def of                     |      |
| General: Use new error commands . . . . .     | 760  | \f@warn@break . . . . .                     | 591  |
| 1994-05-21 ltsect.dtx v1.0d                   |      | 1994-05-23 ltoutput.dtx v1.0u               |      |
| General: Use new error commands . . . . .     | 898  | \activechar@info: Added                     |      |
| 1994-05-21 ltab.dtx v1.0f                     |      | \MessageBreak . . . . .                     | 1114 |
| General: Use new error commands . . . . .     | 837  | \@writesetup: Changed resetting of          |      |
| 1994-05-21 ltxref.dtx v1.1b                   |      | \protect after shipout to use               |      |
| General: Use new warning commands . . . . .   | 738  | \aftergroup . . . . .                       | 1114 |
| \newlabel: Use new warning                    |      | General: Added \MessageBreak. . . . .       | 1087 |
| commands . . . . .                            | 741  | Changed resetting of \protect after         |      |
| 1994-05-22 ltclass.dtx v1.0f                  |      | shipout. . . . .                            | 1087 |
| General: Use new warning and error            |      | 1994-05-24 lterror.dtx v1.2e                |      |
| commands . . . . .                            | 974  | \@latex@info@no@line: Macro added . . . . . | 392  |
| 1994-05-22 ltdefns.dtx v1.1f                  |      | 1994-05-24 lterror.dtx v1.2f                |      |
| General: Use new warning and error            |      | General: (DPC) wrap long lines . . . . .    | 388  |
| cmds . . . . .                                | 80   | 1994-05-24 lfntcmd.dtx v3.3i                |      |
| 1994-05-22 lterror.dtx v1.1e                  |      | General: Tidying and typos fixed . . . . .  | 686  |
| General: (ASAJ) Replaced bgroup by            |      | 1994-05-24 ltmiscen.dtx v1.0q               |      |
| begingroup in error messages, to              |      | \@currenvline: Use \empty as outer          |      |
| stop extra mathords creeping into             |      | default . . . . .                           | 773  |
| math mode. . . . .                            | 388  |                                             |      |

|                                     |                                                                                                 |                                                                                                                                                    |            |
|-------------------------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1994-05-25 ltdirchk.dtx v1.0g       |                                                                                                 | \framebox: New version, so \width is correct in \framebox . . . . .                                                                                | 826        |
|                                     | \filename@parse: Mac parser had "typo for : . . . . .                                           | 12                                                                                                                                                 |            |
| 1994-05-25 ltfnctcmd.dtx v3.3j      |                                                                                                 | \LaTeX: Add \m@th to force math size calculations . . . . .                                                                                        | 441        |
|                                     | General: Insertion of \aftergroups to implement \nocorr moved to the end of the group . . . . . | 686                                                                                                                                                |            |
|                                     | \check@icr: Macros added . . . . .                                                              | 690                                                                                                                                                |            |
|                                     | \check@nocorr@: Insertion of \aftergroups moved and defaults set up for efficiency . . . . .    | 690                                                                                                                                                |            |
|                                     | \DeclareTextFontCommand: \expandafter inserted . . . . .                                        | 688                                                                                                                                                |            |
|                                     | Insertion of \aftergroups moved . . . . .                                                       | 688                                                                                                                                                |            |
| 1994-05-25 ltoutput.dtx v1.0v       |                                                                                                 | General: Extra documentation. . . . .                                                                                                              | 1087       |
| 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. . . . . | 909        |
| 1994-05-25 ltthm.dtx v1.0c          |                                                                                                 | General: Modify documentation . . . . .                                                                                                            | 894        |
| 1994-05-25 ltvers.dtx v1.0d         |                                                                                                 | General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here) . . . . .                                                            | 37         |
| 1994-05-25 ltxref.dtx v1.1c         |                                                                                                 | General: Modify documentation . . . . .                                                                                                            | 738        |
| 1994-05-26 ltfiles.dtx LaTeX2e      |                                                                                                 | \missingfileerror: Modify message format . . . . .                                                                                                 | 461        |
| 1994-05-26 ltlogos.dtx v1.1c        |                                                                                                 | General: Remove \SLiTeX logo . . . . .                                                                                                             | 441        |
| 1994-05-26 ltmiscen.dtx v1.0r       |                                                                                                 | General: \literal removed . . . . .                                                                                                                | 784        |
| 1994-05-26 ltplain.dtx v1.1m        |                                                                                                 | \iterate: (CAR) added \long . . . . .                                                                                                              | 29         |
|                                     | \underbar: (CAR/FMi) changed to use box \tw@ . . . . .                                          | 31                                                                                                                                                 |            |
| 1994-05-26 ltplain.dtx v1.1p        |                                                                                                 | \underbar: (DPC) changed to use \sbox . . . . .                                                                                                    | 31         |
| 1994-05-29 ltssdcl.dtx v2.1j        |                                                                                                 | General: Use new error commands . . . . .                                                                                                          | 602        |
| 1994-05-31 ltfinal.dtx v1.0n        |                                                                                                 | General: Renamed lthyphen.* to lthyphen.*. . . . .                                                                                                 | 1174       |
| 1994-06-01 ltboxes.dtx v1.0i        |                                                                                                 | \framebox: Macro added. . . . .                                                                                                                    | 827        |
|                                     | \ifframebox: New version, so \width is correct in \framebox . . . . .                           | 826                                                                                                                                                |            |
|                                     | \fbox: New version, using \framebox . . . . .                                                   | 825                                                                                                                                                |            |
|                                     | \@framebox . . . . .                                                                            | 825                                                                                                                                                |            |
| 1994-06-01 ltlogos.dtx v1.1d        |                                                                                                 | \LaTeX: Add \m@th to force math size calculations . . . . .                                                                                        | 441        |
| 1994-06-01 ltoutput.dtx v1.0w       |                                                                                                 | General: Tidied up typesetting. . . . .                                                                                                            | 1087       |
| 1994-06-08 ltfinal.dtx v1.0m        |                                                                                                 | General: Add patch file system . . . . .                                                                                                           | 1190       |
| 1994-06-09 ltfinal.dtx v1.0n        |                                                                                                 | General: For TeX2, do not set codes for higher half of character table. . . . .                                                                    | 1179, 1187 |
| 1994-06-09 ltfnctcmd.dtx v3.3k      |                                                                                                 | General: Tidying and typos fixed in documentation . . . . .                                                                                        | 686        |
| 1994-06-18 ltfnctcmd.dtx v3.3l      |                                                                                                 | General: Added check for empty text . . . . .                                                                                                      | 686        |
| 1994-06-22 ltfnctcmd.dtx v3.3m      |                                                                                                 | \check@nocorr@: Added check for empty text . . . . .                                                                                               | 690        |
|                                     | General: Removed space from \nfss@text . . . . .                                                | 686                                                                                                                                                |            |
|                                     | Renamed \check@nocorr . . . . .                                                                 | 686                                                                                                                                                |            |
|                                     | \check@nocorr@: Renamed \check@nocorr to \text@command to improve \long error message . . . . . | 690                                                                                                                                                |            |
|                                     | \DeclareTextFontCommand: Removed space from \nfss@text . . . . .                                | 688                                                                                                                                                |            |
| 1994-06-22 ltmath.dtx v1.2t classes |                                                                                                 | \mathindent: Set \mathindent at the end of the class instead of at begin document . . . . .                                                        | 798        |
| 1994-07-20 ltlogos.dtx v1.1e        |                                                                                                 | \LaTeX: Save a few tokens . . . . .                                                                                                                | 441        |
|                                     | \LaTeXe: Save a few tokens . . . . .                                                            | 441                                                                                                                                                |            |
| 1994-07-20 ltpage.dtx v1.0h         |                                                                                                 | \sloppy: Save a few tokens . . . . .                                                                                                               | 972        |
| 1994-09-16 ltfsbas.dtx v2.1s        |                                                                                                 | \nfss@catcodes: Reset [ and ] as well, just in case . . . . .                                                                                      | 538        |
| 1994-10-07 ltoutenc.dtx v1.5l       |                                                                                                 | General: Moved the ogonek accent. . . . .                                                                                                          | 466        |
| 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 . . . . .                                                                                                                      | 898        |
| 1994-10-14 fontdef.dtx v2.2a        |                                                                                                 | General: New coding . . . . .                                                                                                                      | 662        |
| 1994-10-14 ltfsini.dtx v2.2a        |                                                                                                 | General: New coding for cfg files . . . . .                                                                                                        | 634        |
| 1994-10-14 ltmiscen.dtx v1.0s       |                                                                                                 | General: Move math to other file . . . . .                                                                                                         | 760        |

|                                                             |     |                                                 |     |
|-------------------------------------------------------------|-----|-------------------------------------------------|-----|
| 1994-10-14 ltplain.dtx v1.1a                                |     | 1994-10-25 fontdef.dtx v2.2c                    |     |
| General: Moved code to other files. . .                     | 14  | General: Added OMSenc.def . . . . .             | 664 |
| 1994-10-15 ltfssbas.dtx v2.1t                               |     | 1994-10-25 ltboxes.dtx v1.0l                    |     |
| <code>\extract@alph@from@version:</code> Warn               |     | <code>\@isavepicbox:</code> missing percent     |     |
| if math alpha is used outside                               |     | (moved from ltpatch) . . . . .                  | 824 |
| math . . . . .                                              | 546 | 1994-10-25 ltdefns.dtx v1.2b                    |     |
| 1994-10-18 ltboxes.dtx v1.0j                                |     | General: Documentation                          |     |
| <code>\@frameb@x: \leavevmode</code> added . . .            | 827 | improvements . . . . .                          | 80  |
| <code>\@iframebox: \leavevmode</code> moved to              |     | 1994-10-25 ltoutenc.dtx 1.6a                    |     |
| <code>\@frameb@x</code> . . . . .                           | 826 | General: Added <code>\textdollar,</code>        |     |
| <code>\@parboxto:</code> Macro added to remove              |     | <code>\textbraceleft, \textbraceright,</code>   |     |
| misuse of <code>\empty</code> . . . . .                     | 828 | <code>\textsterling, \textunderline.</code>     | 484 |
| General: stuff from ltpatch done . . .                      | 819 | Removed <code>\textbraceleft,</code>            |     |
| <code>\fbox: \long</code> added . . . . .                   | 825 | <code>\textbraceright, \textunderline</code> to |     |
| <code>\mbox: \long</code> added . . . . .                   | 820 | give them their proper names. . .               | 484 |
| <code>\sbox: \long</code> added . . . . .                   | 824 | 1994-10-25 ltoutenc.dtx v1.6a                   |     |
| 1994-10-18 ltclass.dtx v1.0j                                |     | General: Added                                  |     |
| General: Move <code>\listfiles</code> to                    |     | <code>\ProvideTextCommand,</code>               |     |
| <code>ltfiles.dtx</code> . . . . .                          | 974 | <code>\UseTextSymbol, \UseTextAccent,</code>    |     |
| 1994-10-18 ltdefns.dtx v1.2a                                |     | <code>\DeclareTextSymbolDefault,</code>         |     |
| <code>\@star@or@long:</code> macro added . . .              | 83  | <code>\DeclareTextAccentDefault,</code>         |     |
| General: Add extra test for <code>\endgraf</code>           | 80  | <code>\DeclareTextCommandDefault,</code>        |     |
| Add star-forms for all commands . . .                       | 80  | and                                             |     |
| <code>\renew@environment:</code> reset end                  |     | <code>\ProvideTextCommandDefault.</code> .      | 466 |
| command . . . . .                                           | 87  | Added the <code>\Provide</code> commands,       |     |
| 1994-10-18 ltfiles.dtx v1.0i                                |     | and the default definitions. . . . .            | 470 |
| <code>\listfiles:</code> code moved here from               |     | Added the defaults. . . . .                     | 478 |
| <code>ltclass</code> . . . . .                              | 463 | Added the files OT1enc.def,                     |     |
| 1994-10-18 ltoutenc.dtx v1.5l                               |     | T1enc.def and OMSenc.def. . . . .               | 478 |
| General: Added new definitions of                           |     | Added the OMS encoding. . . . .                 | 490 |
| <code>\patterns</code> and <code>\hyphenation</code> . . .  | 478 | 1994-10-27 ltoutenc.dtx 1.6b                    |     |
| 1994-10-18 ltoutenc.dtx v1.5m                               |     | General: Added <code>\textasciicircum</code>    |     |
| General: Added new definitions of                           |     | <code>\textasciitilde \textbackslash</code>     |     |
| <code>\patterns</code> and <code>\hyphenation</code> . . .  | 466 | <code>\textbar \textbraceleft</code>            |     |
| 1994-10-18 ltsect.dtx v1.0g                                 |     | <code>\textbraceright</code>                    |     |
| <code>\@dottedtocline:</code> Added                         |     | <code>\textcompwordmark \textemdash</code>      |     |
| <code>\normalcolor</code> for page number . . .             | 909 | <code>\textendash \textexclamdown</code>        |     |
| General: Added <code>\normalcolor</code> . . . . .          | 898 | <code>\textgreater \texthypenchar</code>        |     |
| 1994-10-19 ltfssbas.dtx v2.1t                               |     | <code>\texthypen \textless</code>               |     |
| <code>\DeclareFontEncoding:</code> Add missing              |     | <code>\textquestiondown</code>                  |     |
| <code>\relax</code> . . . . .                               | 527 | <code>\textquotedblleft</code>                  |     |
| 1994-10-23 lfsstrc.dtx v23.k                                |     | <code>\textquotedblright</code>                 |     |
| <code>\every@math@size:</code> Renamed to                   |     | <code>\textquotedbl \textquotel</code>          |     |
| <code>\every@math@size</code> . . . . .                     | 581 | <code>\textquoteright</code>                    |     |
| 1994-10-23 ltmath.dtx v1.0l                                 |     | <code>\textunderscore</code>                    |     |
| <code>\eqnnum:</code> Added <code>\normalcolor</code> since |     | <code>\textvisible</code> . . . . .             | 484 |
| <code>\eqno</code> introduces a subgroup of the             |     | Added: <code>\textemdash \textendash</code>     |     |
| displayed math group . . . . .                              | 794 | <code>\textexclamdown</code>                    |     |
| <code>\ensuremath:</code> Remove extra braces:              |     | <code>\texthypenchar \texthypen</code>          |     |
| but see p 168 of Leslie's book . . .                        | 797 | <code>\textquestiondown</code>                  |     |
| 1994-10-24 ltboxes.dtx v1.0k                                |     | <code>\textquotedblleft</code>                  |     |
| <code>\fbox:</code> Inner braces added (to fix              |     | <code>\textquotedblright</code>                 |     |
| latex/1061) . . . . .                                       | 825 | <code>\textquoteright</code> . . . . .          | 482 |

|            |                                                                                                                                                                   |                                                                                                                                          |      |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|------|
| 1994-10-27 | ltoutenc.dtx v1.5d                                                                                                                                                | Made <code>\textless</code> and <code>\textgreater</code> come from OML. . . . .                                                         | 480  |
|            | General: Rewrote                                                                                                                                                  | Moved math commands here from ltmath. . . . .                                                                                            | 482  |
|            | <code>\DeclareTextSymbol</code> to define its argument to use the current encoding by default, to fit with <code>\DeclareTextCommand</code> . . . . .             | Removed <code>\textregistered</code> . . . . .                                                                                           | 480  |
|            | 470                                                                                                                                                               | Rewrote <code>\copyright</code> to use <code>\textcircled</code> . . . . .                                                               | 480  |
| 1994-10-27 | ltoutenc.dtx v1.6b                                                                                                                                                | 1994-10-31 fontdef.dtx v2.2d                                                                                                             |      |
|            | General: Added <code>\textbackslash</code> . . . . .                                                                                                              | General: Added OMLenc.def . . . . .                                                                                                      | 664  |
|            | Added more defaults for OT1. . . . .                                                                                                                              |                                                                                                                                          |      |
|            | Removed the enc.def files . . . . .                                                                                                                               |                                                                                                                                          |      |
|            | Removed the files OT1enc.def, T1enc.def and OMSenc.def. . . . .                                                                                                   | General: ... and moved further down                                                                                                      | 664  |
|            | 478                                                                                                                                                               | 1994-10-31 ltfloor.dtx v1.1a                                                                                                             |      |
|            | Renamed <code>\textlbrace</code> to <code>\textbraceleft</code> and <code>\textrbrace</code> to <code>\textbraceright</code> . . . . .                            | <code>\@dblfloor</code> : Major changes since two-column and one-column cases merged . . . . .                                           | 914  |
|            | 490                                                                                                                                                               | <code>\@dblfloorset</code> : Macro added . . . . .                                                                                       | 914  |
| 1994-10-29 | ltmath.dtx 1.0m                                                                                                                                                   | Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved . . . . . | 914  |
|            | General: ASAJ: Added                                                                                                                                              | 1994-10-29 \endfloatbox: (DPC/CAR) Extra box added to remove colour resetting from vmode . . . . .                                       | 920  |
|            | <code>\DeclareMathOperator</code> . . . . .                                                                                                                       | <code>\@floatboxreset</code> : Macro added . . . . .                                                                                     | 918  |
|            | 785                                                                                                                                                               | <code>\@footnotetext</code> : (DPC/CAR) Move colour setting to output routine . . . . .                                                  | 930  |
|            | ASAJ: Tidied up documentation. . . . .                                                                                                                            | <code>\@savemarbox</code> : (DPC/CAR) Extra box added for colour . . . . .                                                               | 924  |
| 1994-10-29 | ltmath.dtx v1.0m                                                                                                                                                  | <code>\@setfps</code> : Macro added . . . . .                                                                                            | 915  |
|            | General: ASAJ: Added                                                                                                                                              | <code>\@dblfloat</code> : Macros removed: <code>\@dbfl, \@x dblfloat</code> . . . . .                                                    | 920  |
|            | <code>\mathellipsis, \mathdollar</code> and <code>\mathsterling</code> . . . . .                                                                                  | <code>\@xfloat</code> : (DPC/CAR) Extra box added to remove colour resetting from vmode . . . . .                                        | 916  |
|            | 792                                                                                                                                                               | Major changes, removing setting of local variables, space hacks etc; two-column and one-column cases merged . . . . .                    | 915  |
|            | ASAJ: Removed <code>\dag, \ddag</code> . . . . .                                                                                                                  | Reset hook added . . . . .                                                                                                               | 916  |
|            | ASAJ: Renamed <code>\S</code> and <code>\P</code> to <code>\mathsection</code> and <code>\mathparagraph</code> and made them <code>\mathchardef</code> s. . . . . | 1994-10-31 \xympar: (DPC/CAR) Extra box added since needed for floats . . . . .                                                          | 925  |
|            | 792                                                                                                                                                               | <code>\@xmpar</code> : (DPC/CAR) Extra box added to remove colour resetting from vmode . . . . .                                         | 916  |
| 1994-10-29 | ltoutenc.dtx v1.6c                                                                                                                                                | <code>\@fps@dbl</code> : Macro added . . . . .                                                                                           | 915  |
|            | General: Added commands like <code>\dots</code> for use in text and math. . . . .                                                                                 | 1994-10-31 \output: (DPC/CAR) Colour resetting moved to here . . . . .                                                                   | 1111 |
|            | 478                                                                                                                                                               | <code>\@makecol</code> : (DPC/CAR) Colour resetting moved to here . . . . .                                                              | 1111 |
|            | Renamed <code>\P, \S, \dag</code> and <code>\ddag</code> to <code>\textparagraph, \textsection, \textdagger</code> and <code>\textdaggerdbl</code> . . . . .      | <code>\@topnewpage</code> : (DPC/CAR) Extra box added to remove colour resetting from vmode . . . . .                                    | 1103 |
|            | 466                                                                                                                                                               | (DPC/CAR) Use <code>\color@begingroup</code> for colour . . . . .                                                                        | 1103 |
| 1994-10-30 | ltdefns.dtx v1.2c                                                                                                                                                 | (DPC/CAR) Use <code>\normalcolor</code> . . . . .                                                                                        | 1103 |
|            | <code>\@onelvel@sanitize</code> : Macro added . . . . .                                                                                                           | 1994-11-02 ltoutenc.dtx v1.6d                                                                                                            |      |
|            | General: (CAR) <code>\@onelvel@sanitize</code> added . . . . .                                                                                                    | General: Wrapped lines longer than 70 characters. . . . .                                                                                | 466  |
| 1994-10-30 | ltdefns.dtx v1.2f                                                                                                                                                 |                                                                                                                                          |      |
|            | General: (DPC) <code>\newwrite</code> 's moved to ltfiles . . . . .                                                                                               |                                                                                                                                          |      |
|            | 80                                                                                                                                                                |                                                                                                                                          |      |
| 1994-10-30 | ltmath.dtx v1.0n                                                                                                                                                  |                                                                                                                                          |      |
|            | General: ASAJ: Moved the new commands to ltoutenc. . . . .                                                                                                        |                                                                                                                                          |      |
|            | 792                                                                                                                                                               |                                                                                                                                          |      |
| 1994-10-30 | ltoutenc.dtx v1.6d                                                                                                                                                |                                                                                                                                          |      |
|            | General: Added                                                                                                                                                    |                                                                                                                                          |      |
|            | <code>\DeclareTextCompositeCommand</code> . . . . .                                                                                                               |                                                                                                                                          |      |
|            | 466                                                                                                                                                               |                                                                                                                                          |      |
|            | Added <code>\textcircled</code> . . . . .                                                                                                                         |                                                                                                                                          |      |
|            | 466, 480, 490                                                                                                                                                     |                                                                                                                                          |      |
|            | Added <code>\t</code> . . . . .                                                                                                                                   |                                                                                                                                          |      |
|            | 480                                                                                                                                                               |                                                                                                                                          |      |
|            | Added math commands. . . . .                                                                                                                                      |                                                                                                                                          |      |
|            | 466                                                                                                                                                               |                                                                                                                                          |      |
|            | Added OML encoding. . . . .                                                                                                                                       |                                                                                                                                          |      |
|            | 466, 480                                                                                                                                                          |                                                                                                                                          |      |
|            | Added the OML encoding. . . . .                                                                                                                                   |                                                                                                                                          |      |
|            | 491                                                                                                                                                               |                                                                                                                                          |      |

|                                                                                                                 |  |                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------|--|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1994-11-03 ltclass.dtx v1.0k                                                                                    |  | 1994-11-04 ltpage.dtx v1.0e                                                                                                                       |
| General: Move \@missingfileerror<br>to ltfiles ..... 978                                                        |  | \markright: Added<br>\unexpandable@protect. ASAJ. 970                                                                                             |
| 1994-11-03 ltdirchk.dtx v1.0i                                                                                   |  | 1994-11-04 ltsect.dtx 1.0h                                                                                                                        |
| General: Generate an error if latex.ltx<br>not used with clean initex ..... 1                                   |  | \@sect: (ASAJ) Added<br>\protected@edef. ..... 902                                                                                                |
| 1994-11-03 ltfiles.dtx v1.0j                                                                                    |  | General: (ASAJ) Added<br>\protected@xdef to \thanks. ... 898                                                                                      |
| \@missingfileerror: Move here from<br>ltclass ..... 461                                                         |  | 1994-11-04 ltsect.dtx v1.0h                                                                                                                       |
| 1994-11-04 ltboxes.dtx v1.0m                                                                                    |  | General: Added \protected@write to<br>\addtocontents. ASAJ. ..... 908                                                                             |
| \@mpfootnotetext: Added<br>\protected@edef. ASAJ. .... 832                                                      |  | \addcontentsline: Added<br>\protected@write to<br>\addcontentsline. ASAJ. .... 907                                                                |
| 1994-11-04 ltdefns.dtx v1.2e                                                                                    |  | 1994-11-04 ltab.dtx v1.0h                                                                                                                         |
| General: Added<br>\set@display@protect to<br>\typeout. ASAJ. ..... 80                                           |  | \@mkpream: (ASAJ) Added<br>\unexpandable@protect to<br>\@mkpream. ..... 856                                                                       |
| Added commands for setting and<br>restoring \protect. ASAJ. ..... 92                                            |  | \multicolumn: (ASAJ) added<br>\set@typeset@protect. .... 852                                                                                      |
| Rewrote protected short commands<br>using \x@protect. ASAJ. ..... 90                                            |  | 1994-11-04 ltxref.dtx v1.1d                                                                                                                       |
| 1994-11-04 lterror.dtx v1.2g                                                                                    |  | \label: (ASAJ) Added<br>\protected@edef ..... 742                                                                                                 |
| General: Added<br>\set@display@protect to<br>\Generic* commands. ASAJ. ... 388                                  |  | (ASAJ) Added \protected@write 742                                                                                                                 |
| 1994-11-04 ltfiles.dtx v1.0k                                                                                    |  | 1994-11-05 ltboxes.dtx v1.0n                                                                                                                      |
| \nofiles: Added setting of<br>\protected@write, \makeindex<br>and \makeglossary to \nofiles.<br>ASAJ. ..... 449 |  | \@mpfootnotetext: Color resetting for<br>footnotes moved to endminipage:<br>as for main page. ..... 832                                           |
| \protected@write: Macro added<br>ASAJ. ..... 449                                                                |  | \color@endbox: macro added for color<br>support ..... 822                                                                                         |
| 1994-11-04 ltfloat.dtx v1.1b                                                                                    |  | \color@hbox: macro added for color<br>support ..... 822                                                                                           |
| \@footnotetext: (ASAJ) Added<br>\protected@edef. ..... 930                                                      |  | \endminipage: Color resetting for<br>footnotes moved to here: as for<br>main page. ..... 832                                                      |
| \ffootnotemark: Added<br>\protected@xdef to<br>\ffootnotemark. ..... 931                                        |  | 1994-11-05 ltboxes.dtx v1.0o                                                                                                                      |
| 1994-11-04 ltidxglo.dtx v1.1b                                                                                   |  | \@mpfootnotetext: Color groups<br>restored here. ..... 832                                                                                        |
| \@wrglossary: Added<br>\protected@write to<br>\@wrglossary. ..... 936                                           |  | 1994-11-05 ltfloat.dtx v1.1c                                                                                                                      |
| \@wrindex: Added \protected@write<br>to \@wrindex. ..... 935                                                    |  | \@dblflset: Add compatibility with<br>old version of \@xfloat. ..... 914                                                                          |
| General: Removed \if@filesw from<br>\makeindex. ..... 934                                                       |  | \@endfloatbox: Use new \color@hbox<br>concept. ..... 920                                                                                          |
| \makeglossary: Removed \if@filesw<br>from \makeglossary. ..... 935                                              |  | \@footnotetext: Removed<br>\normalcolor (again) ..... 930                                                                                         |
| 1994-11-04 ltmiscen.dtx v1.0t                                                                                   |  | \@savemarbox: Use new \color@hbox<br>concept. ..... 924                                                                                           |
| \@writefile: Removed setting of<br>\protect. ASAJ. ..... 767                                                    |  | \@setfps: Add compatibility with old<br>version of \@xfloat. ..... 915                                                                            |
| 1994-11-04 ltoutenc.dtx v1.6f                                                                                   |  | \@xfloat: Add compatibility with old<br>version of \@xfloat: but the<br>arguments, provided at exorbitant<br>cost, are now completely ignored 915 |
| General: Added \_ ..... 481                                                                                     |  | Use new \color@hbox concept. ... 916                                                                                                              |
| Added \mathunderscore. ..... 482                                                                                |  |                                                                                                                                                   |

|                                                                                                                   |      |                                                                                                                                        |      |
|-------------------------------------------------------------------------------------------------------------------|------|----------------------------------------------------------------------------------------------------------------------------------------|------|
| \@xympar: Use new \color@hbox concept.                                                                            | 925  | (DPC) reduce save stack usage<br>latex/1742                                                                                            | 549  |
| 1994-11-05 ltoutenc.dtx v1.6g<br>General: Added setting of<br>\@typeset@protect to \patterns<br>and \hyphenation. | 478  | 1994-11-10 ltbibl.dtx v1.1c<br>General: Fix \nocite{*}                                                                                 | 937  |
| 1994-11-05 ltoutput.dtx v1.1b<br>\@topnewpage: Use new \color@hbox concept.                                       | 1103 | \nocite: Fix \nocite{*}                                                                                                                | 940  |
| \@writesetup: Change protect<br>settings for new-style, protect-free<br>aux-files.                                | 1114 | 1994-11-10 ltmath.dtx v1.2v classes<br>eqnarray: Added value of \parskip to<br>\abovedisplayskip to compensate<br>for negative \topsep | 800  |
| 1994-11-05 ltoutput.dtx v1.1c<br>\@begindvi: Added macro                                                          | 1118 | 1994-11-10 ltoutput.dtx v1.1e<br>\@writesetup: Modify \protect<br>setting                                                              | 1114 |
| \@begindvibox: Added macro                                                                                        | 1099 | 1994-11-10 lplain.dtx v1.1b<br>General: (CAR) added patch to \loop.                                                                    | 14   |
| \@writesetup: Add new \AtBeginDvi<br>concept                                                                      | 1114 | \iterate: (CAR) added extra \relax                                                                                                     | 29   |
| \AtBeginDvi: Added macro                                                                                          | 1099 | 1994-11-11 lspace.dtx v1.2a<br>\: (DPC) Make robust                                                                                    | 425  |
| 1994-11-06 ltfsbas.dtx v2.1u<br>\cf@encoding: New macro                                                           | 533  | 1994-11-12 lfntcmd.dtx v3.3o<br>\normalsize: Added \MessageBreak                                                                       | 694  |
| \DeclareFixedFont: Renamed<br>\every@size to<br>\every@math@size                                                  | 525  | 1994-11-12 llists.dtx v1.2b lspace<br>\endtrivlist: Changed order of tests<br>to make \noitemerror correct:<br>end of an era.          | 810  |
| 1994-11-06 ltfsini.dtx v2.2b<br>\@setsizes: Use \@typeset@protect                                                 | 656  | 1994-11-12 ltmiscen.dtx v1.0u<br>center: Changed end macro to \def:<br>safer and consistent                                            | 775  |
| 1994-11-06 lfsstrc.dtx v2.3k<br>\glb@currsize: New implementation                                                 | 580  | flushleft: Changed end macro to<br>\def: safer and consistent                                                                          | 776  |
| \try@simples: New implementation                                                                                  | 591  | flushright: Changed end macro to<br>\def: safer and consistent                                                                         | 777  |
| \try@size@substitution: New<br>implementation                                                                     | 591  | 1994-11-12 lplain.dtx v1.1c<br>General: Comment out more encoding<br>specific commands                                                 | 31   |
| \tryis@simple: New implementation                                                                                 | 592  | 1994-11-12 lspace.dtx v1.2b<br>\addpenalty: Corrected error<br>message                                                                 | 433  |
| 1994-11-07 fontdef.dtx v2.2f<br>General: (DPC) Add<br>\DeclareMathSizes declarations                              | 669  | \addvspace: Corrected error message                                                                                                    | 432  |
| (DPC) Updated to use<br>\ProvidesFile                                                                             | 664  | 1994-11-13 lspace.dtx v1.2c<br>\addpenalty: Recorrected error<br>message                                                               | 433  |
| 1994-11-07 ltfiles.dtx v1.0m<br>\document: Renamed \every@size to<br>\every@math@size                             | 445  | \addvspace: Recorrected error<br>message                                                                                               | 432  |
| 1994-11-07 lplain.dtx v1.0l<br>\@unused: move here from ltdefns,<br>remove duplicate \mainaux                     | 23   | 1994-11-14 ltoutput.dtx v1.1f<br>\@begindvi: Use normal box register:<br>why a box?                                                    | 1118 |
| 1994-11-07 preload.dtx v2.1e<br>General: (DPC) Updated to use<br>\ProvidesFile                                    | 683  | \@begindvibox: Use normal box<br>register: why a box?                                                                                  | 1099 |
| 1994-11-09 ltboxes.dtx v1.0p<br>\@finalstrut: Revert \finalstrut to<br>2.09 equivalent (from ltpatch)             | 835  | \@writesetup: Modify new<br>\AtBeginDvi concept                                                                                        | 1114 |
| General: more color changes...                                                                                    | 819  | General: Removed old definition of<br>\@testfp.                                                                                        | 1087 |
| 1994-11-09 ltfsbas.dtx v2.1v<br>\vpt: (DPC) macros added, from<br>setsizes.dtx                                    | 549  | \\: (DPC) Macro modified                                                                                                               | 425  |

|                                                                                         |                                                                                                                              |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 1994-11-14 lttab.dtx v1.0i<br>\tabularnewline: (DPC) Macro<br>added ..... 851           | 1994-11-18 ltfsdcl.dtx v2.1m<br>\DeclareMathDelimiter: (DPC)<br>\xpandafter instead of \next .. 626                          |
| 1994-11-16 fontdef.dtx v2.2h<br>General: (DPC) Removed \{ and \} .. 664                 | 1994-11-18 ltfssrc.dtx v2.3m<br>General: \next to \reserved@f ... 572                                                        |
| 1994-11-17 ltboxes.dtx v1.0q<br>General: \tempa to \reserved@a .. 819                   | 1994-11-18 ltmath.dtx v1.0p<br>\phantom: (DPC) colour support ... 787<br>(DPC) use \xpandafter instead of<br>\next ..... 787 |
| 1994-11-17 ltclass.dtx v1.0l<br>General: \tempa to \reserved@a .. 974                   | \prime@s: (DPC) use \let@token<br>instead of \next and<br>\xpandafter instead of \nxt .. 792                                 |
| 1994-11-17 ltcntrl.dtx v1.0b<br>General: \tempa to \reserved@a .. 384                   | \smash: (DPC) colour support .... 788<br>(DPC) use \xpandafter instead of<br>\next ..... 788                                 |
| 1994-11-17 ltdefns.dtx v1.0g<br>General: \tempa to \reserved@a .. 80                    | 1994-11-21 ltfloat.dtx v1.1f<br>\endfloatbox: Added reset of<br>minipage flag ..... 920                                      |
| 1994-11-17 ltdirchk.dtx v1.0j<br>General: \tempa to \reserved@a .... 1                  | Corrected position of<br>\nouter@nobreak ..... 920                                                                           |
| 1994-11-17 lterror.dtx v1.2h<br>General: \tempa to \reserved@a .. 388                   | \marginparreset: Macro added ... 924                                                                                         |
| 1994-11-17 ltfiles.dtx v1.0n<br>General: \tempa to \reserved@a .. 442                   | \savemarbox: Added \setminipage<br>etc ..... 924                                                                             |
| 1994-11-17 ltfinal.dtx v1.0o<br>General: \tempa to \reserved@a .. 1174                  | Added resetting of size and font .. 924                                                                                      |
| 1994-11-17 ltfloat.dtx v1.1e<br>General: \tempa to \reserved@a .. 911                   | Changed to \color@vbox ..... 924                                                                                             |
| 1994-11-17 ltftntcmd.dtx v3.3p<br>General: \tempa to \reserved@a .. 686                 | Use \setnobreak etc ..... 924                                                                                                |
| 1994-11-17 ltfsbas.dtx v2.1w<br>General: \tempa to \reserved@a .. 523                   | \setminipage: Macro added ..... 918                                                                                          |
| 1994-11-17 ltfsdcl.dtx v2.1m<br>General: \tempa to \reserved@a .. 602                   | \setnobreak: Macro added ..... 918                                                                                           |
| 1994-11-17 ltfssrc.dtx v2.3l<br>General: \tempa to \reserved@a .. 572                   | \xfloat: Added \setminipage ... 916                                                                                          |
| 1994-11-17 ltmath.dtx v1.0o<br>General: \tempa to \reserved@a .. 785                    | Added resetting of size and font .. 916                                                                                      |
| 1994-11-17 ltmiscen.dtx v1.0v<br>General: \tempa to \reserved@a .. 760                  | Changed to \color@vbox so that<br>large floats overflow at the<br>bottom ..... 916                                           |
| 1994-11-17 ltoutenc.dtx v1.6h<br>General: (DPC) \tempa to<br>\reserved@a ..... 466      | Missing percents reinserted after 4,<br>8: these are not numbers. .... 915                                                   |
| 1994-11-17 ltoutput.dtx v1.1h<br>General: \tempa to \reserved@a. 1087                   | Use \setnobreak ..... 916                                                                                                    |
| 1994-11-17 ltpictur.dtx v1.0f<br>General: \tempa to \reserved@a .. 862                  | \xympar: Changed to \color@vbox 925                                                                                          |
| 1994-11-17 ltsect.dtx v1.0i<br>General: \tempa to \reserved@a .. 898                    | 1994-11-21 ltoutput.dtx v1.1i<br>\addtocurcol: Added \if@nobreak<br>test before float box .... 1129, 1132                    |
| 1994-11-17 lttab.dtx v1.0j<br>General: \tempa to \reserved@a .. 837                     | \specialoutput: Added \if@nobreak<br>test ..... 1107                                                                         |
| 1994-11-18 ltboxes.dtx v1.0r<br>\color@vbox: macro added for color<br>support ..... 822 | \topnewpage: Changed to<br>\color@vbox ..... 1103                                                                            |
| 1994-11-18 ltfinal.dtx v1.0n<br>General: re-allow slots 127–255 ... 1186                | 1994-11-22 ltfsdcl.dtx v2.1o<br>General: wrap long lines ..... 602                                                           |
| 1994-11-18 ltfsbas.dtx v2.1x<br>General: (DPC) use \reserved@f not<br>\next ..... 523   | 1994-11-22 ltoutenc.dtx v1.6i<br>General: Corrected \dots so that<br>there's no kerning in monowidth<br>fonts. ..... 466     |
|                                                                                         | Corrected typo with<br>\mathunderscore. ..... 466                                                                            |
|                                                                                         | Fixed empty accents. Again. .... 466                                                                                         |

|                                                                                                                                                                |      |                                                  |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------|-----|
| 1994-11-24 ltdefns.dtx v1.2h                                                                                                                                   |      | argument containing lots of<br>commands. . . . . | 472 |
| \@newenv: Added test for \endgraf . . . . .                                                                                                                    | 87   |                                                  |     |
| 1994-11-25 lplain.dtx v1.1f                                                                                                                                    |      |                                                  |     |
| General: (DPC) Comment out lots of<br>obsolete code . . . . .                                                                                                  | 14   |                                                  |     |
| 1994-11-26 lfloat.dtx v1.1b                                                                                                                                    |      |                                                  |     |
| \f footnote: (ASAJ) Added<br>\protected@xdef. . . . .                                                                                                          | 930  |                                                  |     |
| 1994-11-28 lcntrl.dtx v1.0c                                                                                                                                    |      |                                                  |     |
| General: Documentation<br>improvements . . . . .                                                                                                               | 384  |                                                  |     |
| 1994-11-30 lfiles.dtx v1.0o                                                                                                                                    |      |                                                  |     |
| \@dofilelist: Macro added . . . . .                                                                                                                            | 465  |                                                  |     |
| \listfiles: Use \@dofilelist . . . . .                                                                                                                         | 463  |                                                  |     |
| \nofiles: There is no<br>\@gobblethree. . . . .                                                                                                                | 449  |                                                  |     |
| 1994-11-30 ltfssbas.dtx v2.1y                                                                                                                                  |      |                                                  |     |
| \f fontshape: Use \@current@cmd in<br>\@enc@update. ASAJ. . . . .                                                                                              | 532  |                                                  |     |
| 1994-11-30 ltmath.dtx 1.0q                                                                                                                                     |      |                                                  |     |
| General: ASAJ:<br>\DeclareMathOperator moved to<br>AMSLATEX. . . . .                                                                                           | 785  |                                                  |     |
| 1994-11-30 ltmiscen.dtx v1.0w                                                                                                                                  |      |                                                  |     |
| \enddocument@kernel@warnings:<br>(DPC) Do warnings even for<br>\nofiles . . . . .                                                                              | 763  |                                                  |     |
| \enddocument: (DPC) Use<br>\@dofilelist . . . . .                                                                                                              | 762  |                                                  |     |
| 1994-11-30 ltoutenc.dtx 1.7a                                                                                                                                   |      |                                                  |     |
| General: Redefined \a for the new<br>scheme. . . . .                                                                                                           | 478  |                                                  |     |
| 1994-11-30 ltoutenc.dtx v1.6g                                                                                                                                  |      |                                                  |     |
| General: Removed new definitions of<br>\patterns and \hyphenation,<br>since encoding-specific commands<br>now expand in the mouth. . . . .                     | 478  |                                                  |     |
| 1994-11-30 ltoutenc.dtx v1.7a                                                                                                                                  |      |                                                  |     |
| General: Added new code for<br>encoding-specific commands.<br>These now expand in the mouth,<br>which means that ligaturing and<br>kerning can happen. . . . . | 466  |                                                  |     |
| Always load the enc.def file, so that<br>the default encoding for the<br>commands will change. . . . .                                                         | 509  |                                                  |     |
| Redefined \@changed@cmd to expand<br>in the mouth. . . . .                                                                                                     | 470  |                                                  |     |
| Removed \@changed@x@mouth since<br>\@changed@x now expands in the<br>mouth. . . . .                                                                            | 470  |                                                  |     |
| Rewrote \@text@composite so it<br>allows an empty argument, or an                                                                                              |      |                                                  |     |
| 1994-12-01 lfinal.dtx v1.0p                                                                                                                                    |      |                                                  |     |
| General: Renamed lthyphen.* to<br>hyphen.*. . . . .                                                                                                            | 1174 |                                                  |     |
| 1994-12-01 lthyphen.dtx v1.0g                                                                                                                                  |      |                                                  |     |
| General: Rename lthyphen.ltx/cfg to<br>hyphen.ltx/cfg . . . . .                                                                                                | 1172 |                                                  |     |
| 1994-12-01 lplain.dtx v1.1g                                                                                                                                    |      |                                                  |     |
| General: (DPC) More doc changes . . . . .                                                                                                                      | 14   |                                                  |     |
| 1994-12-02 fontdef.dtx v2.2i                                                                                                                                   |      |                                                  |     |
| General: Commented out \ldots.<br>ASAJ. . . . .                                                                                                                | 662  |                                                  |     |
| 1994-12-02 ltfssini.dtx v2.2c                                                                                                                                  |      |                                                  |     |
| \copyright: \copyright is now in<br>ltoutenc. ASAJ . . . . .                                                                                                   | 657  |                                                  |     |
| 1994-12-02 ltlists.dtx v1.0e                                                                                                                                   |      |                                                  |     |
| \@trivlist: RmS: Added check for<br>looping . . . . .                                                                                                          | 809  |                                                  |     |
| 1994-12-02 ltoutenc.dtx 1.7b                                                                                                                                   |      |                                                  |     |
| General: Redefined \a properly. . . . .                                                                                                                        | 478  |                                                  |     |
| 1994-12-02 ltoutenc.dtx v1.7b                                                                                                                                  |      |                                                  |     |
| General: Fixed a bug with \a. . . . .                                                                                                                          | 466  |                                                  |     |
| 1994-12-04 lthyphen.dtx v1.0h                                                                                                                                  |      |                                                  |     |
| General: Documentation edits for<br>/1989 . . . . .                                                                                                            | 1172 |                                                  |     |
| 1994-12-05 ltoutenc.dtx v1.7c                                                                                                                                  |      |                                                  |     |
| General: Added braces to<br>\textcircled. . . . .                                                                                                              | 466  |                                                  |     |
| 1994-12-06 ltfssbas.dtx v2.1z                                                                                                                                  |      |                                                  |     |
| \DeclareFontEncoding: use<br>\nfss@catcodes . . . . .                                                                                                          | 527  |                                                  |     |
| \nfss@catcodes: Added tab char as<br>well . . . . .                                                                                                            | 538  |                                                  |     |
| 1994-12-08 ltoutenc.dtx v1.7d                                                                                                                                  |      |                                                  |     |
| General: Added \null and \sh@ft to<br>\nb and \d. . . . .                                                                                                      | 466  |                                                  |     |
| 1994-12-08 ltab.dtx v1.0k                                                                                                                                      |      |                                                  |     |
| \array: Add \tabularnewline . . . . .                                                                                                                          | 850  |                                                  |     |
| \tabularnewline: (DPC) Made it<br>\relax . . . . .                                                                                                             | 851  |                                                  |     |
| 1994-12-09 ltbibl.dtx v1.1d                                                                                                                                    |      |                                                  |     |
| \bibliographystyle: (DPC) Allow<br>use in preamble. . . . .                                                                                                    | 940  |                                                  |     |
| 1994-12-10 lfloat.dtx v1.1g                                                                                                                                    |      |                                                  |     |
| \@dblfloat: Old version reinstated<br>temporarily . . . . .                                                                                                    | 914  |                                                  |     |
| \@dblfloat: Macro removed<br>temporarily . . . . .                                                                                                             | 914  |                                                  |     |
| Old version reinstated temporarily . . . . .                                                                                                                   | 914  |                                                  |     |
| \@setfps: Macro removed<br>temporarily . . . . .                                                                                                               | 915  |                                                  |     |
| \@dblfloat: Macros reinserted<br>temporarily . . . . .                                                                                                         | 920  |                                                  |     |

|                                                                                                                                                           |     |                                                                  |                                                                           |      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------------------------------------------------------------|---------------------------------------------------------------------------|------|
| \@xfloat: Old version reinstated temporarily . . . . .                                                                                                    | 915 | 1995-04-21 ltclass.dtx v1.0m                                     | \DeclareOption*: Made long /1498                                          | 991  |
| Sanitization added temporarily . . . . .                                                                                                                  | 915 | \endfilecontents: Close input check stream: latex/1487 . . . . . | 1010                                                                      |      |
| General: Some temps reinserted temporarily . . . . .                                                                                                      | 911 | 1995-04-21 ltfinal.dtx v1.0q                                     | General: Allow initial patch level 0                                      | 1190 |
| \fps@dbl: Macro removed temporarily . . . . .                                                                                                             | 915 | 1995-04-21 ltoutenc.dtx v1.7h                                    | General: Added \null \k latex/1274                                        | 466  |
| 1994-12-10 lfntcmd.dtx v3.3q                                                                                                                              |     | 1995-04-22 ltfiles.dtx v1.0p                                     | \includeonly: Allow blanks in argument . . . . .                          | 450  |
| \@@math@egroup: Don't read arguments . . . . .                                                                                                            | 693 | 1995-04-22 ltmiscen.dtx v1.0x                                    | General: Removed extra def of \gobble . . . . .                           | 760  |
| \check@nocorr@: Use \space command for comparison . . . . .                                                                                               | 690 | 1995-04-23 ltsect.dtx v1.0j                                      | \addcontentsline: Use \contentsline internally. . . . .                   | 907  |
| 1994-12-10 ltssdcl.dtx v2.1p                                                                                                                              |     | 1995-04-24 ltbibl.dtx v1.1e                                      | \@citex: Add \mbox to undefined case: latex/1239. . . . .                 | 939  |
| \document@select@group: Surround with braces (add fourth arg) . . . . .                                                                                   | 608 | 1995-04-24 ltbibl.dtx v1.1f                                      | \bibcite: Make \onlypreamble /1388. . . . .                               | 938  |
| \select@group: Surround with braces (add fourth arg) . . . . .                                                                                            | 605 | 1995-04-24 ltcntrl.dtx v1.0d                                     | \@for: Don't expand second argument with \edef: /1317 (DPC) . . . . .     | 387  |
| 1994-12-10 ltoutenc.dtx v1.7e                                                                                                                             |     | 1995-04-24 ltoutput.dtx v1.1j                                    | \fl@tracemessage: Do not add to kernel unless 'trace' specified . . . . . | 1146 |
| General: Added documentation for the OML encoding. . . . .                                                                                                | 466 | 1995-04-24 ltoutput.dtx v1.1l                                    | \begindvibox: Add \vbox latex/1392 . . . . .                              | 1099 |
| Replaced width with \width and ditto height in vrules. . . . .                                                                                            | 466 | 1995-04-24 ltpage.dtx v1.0f                                      | \writesetup: Reset \\ latex/1451 (DPC) . . . . .                          | 1115 |
| 1994-12-14 ltoutenc.dtx v1.7f                                                                                                                             |     | 1995-04-24 ltpage.dtx v1.0f                                      | \fussy: reset \emergencystretch latex/1344 . . . . .                      | 973  |
| General: Added braces to \copyright so it works unbraced in subscripts. . . . .                                                                           | 466 | 1995-04-24 ltplain.dtx v1.1h                                     | \newlanguage: Remove remaining \outer declarations. . . . .               | 17   |
| Added check for math mode in \changed@cmd. . . . .                                                                                                        | 466 | 1995-04-24 ltxref.dtx v1.1e                                      | \newlabel: Make \onlypreamble for /1388. . . . .                          | 741  |
| Commented out \textasciicircum, \textasciitilde, \textbackslash, \textbar, \textgreater, \texthypenchar, \texthypen and \textless to save memory. . . . . | 466 | 1995-04-25 ltnode.dtx v1.2i                                      | \check@c: Make \long for latex/1346                                       | 88   |
| 1995-01-12 ltmath.dtx v1.2y classes                                                                                                                       |     | 1995-04-25 ltnode.dtx v1.2i                                      | \newenvironment: Parse arguments slowly but safely /1507 . . . . .        | 86   |
| \eqnnum: Added \normalcolor . . . . .                                                                                                                     | 798 | 1995-04-25 ltfiles.dtx v1.0q                                     | \document: Removed execution of \every@size latex/1407 . . . . .          | 445  |
| 1995-03-03 ltoutenc.dtx 1.7g                                                                                                                              |     | 1995-04-25 ltsect.dtx v1.0k                                      | \dottedtocline: Added \hbox around dots. . . . .                          | 909  |
| General: Corrected an error in documentation referring to the tabular rather than the tabbing environment. . . . .                                        | 478 |                                                                  |                                                                           |      |
| 1995-04-02 lfntcmd.dtx v3.3r                                                                                                                              |     |                                                                  |                                                                           |      |
| \@@math@egroup: Read them again to be able to add \relax. . . . .                                                                                         | 693 |                                                                  |                                                                           |      |
| 1995-04-02 ltssdcl.dtx v2.1q                                                                                                                              |     |                                                                  |                                                                           |      |
| \document@select@group: fix problem for pr/1275 . . . . .                                                                                                 | 608 |                                                                  |                                                                           |      |
| \select@group: fix problem for pr/1275 . . . . .                                                                                                          | 605 |                                                                  |                                                                           |      |
| \set@mathdelimiter: fix pr/1329 . . . . .                                                                                                                 | 628 |                                                                  |                                                                           |      |
| 1995-04-02 ltssini.dtx v2.2d                                                                                                                              |     |                                                                  |                                                                           |      |
| \not@math@alphabet: add \noexpand to second part of message . . . . .                                                                                     | 655 |                                                                  |                                                                           |      |

|                                                                     |      |                                                                     |     |
|---------------------------------------------------------------------|------|---------------------------------------------------------------------|-----|
| 1995-04-27 ltboxes.dtx v1.0s                                        |      | 1995-05-07 ltplain.dtx v1.1j                                        |     |
| \@frameb@x: Move \leavevmode for<br>graphics/1512 .....             | 827  | General: Use \hb@xt@ .....                                          | 14  |
| \@iframebox: Move \leavevmode for<br>graphics/1512 .....            | 826  | General: Use \hb@xt@ .....                                          | 898 |
| \@iirsbox: Move \leavevmode for<br>graphics/1512 .....              | 835  | General: Use \hb@xt@ .....                                          | 837 |
| \@irsbox: Move \leavevmode for<br>graphics/1512 .....               | 835  | General: Use \hb@xt@ .....                                          | 837 |
| \fbox: Move \leavevmode for<br>graphics/1512 .....                  | 825  | \@citex: Use \@firstofone .....                                     | 939 |
| \raisebox: Move \leavevmode for<br>graphics/1512 .....              | 834  | \bibitem: Removed unnecessary<br>braces .....                       | 938 |
| 1995-04-27 ltfiles.dtx v1.0r                                        |      | \nocite: Use \@firstofone .....                                     | 940 |
| \document: Added \global to support<br>groups in hook .....         | 446  | 1995-05-08 ltdefns.dtx v1.2k                                        |     |
| 1995-04-27 ltmiscen.dtx v1.0y                                       |      | \typein: Use \@firstofone .....                                     | 82  |
| \enddocument: \@checkend moved<br>after hook .....                  | 761  | 1995-05-08 ltdefns.dtx v1.2l                                        |     |
| 1995-04-27 ltplain.dtx v1.1i                                        |      | \typein: Remove unnecessary braces                                  | 82  |
| General: Move \hang and<br>\textrm to latex209.def .....            | 31   | Replace \def by \let .....                                          | 82  |
| 1995-04-29 ltcntrl.dtx v1.0e                                        |      | 1995-05-08 lfsstrc.dtx v2.3n                                        |     |
| General: Moved init of \protect to<br>ltdefns.dtx .....             | 387  | \ifnot@nil: Use \@firstofone ...                                    | 585 |
| Removed unused defs for<br>\@setprotect and<br>\@resetprotect ..... | 387  | 1995-05-11 fontdef.dtx v2.2j                                        |     |
| 1995-04-29 ltdefns.dtx v1.2j                                        |      | General: Updates to some plain<br>macros .....                      | 662 |
| \protect: Init \protect here .....                                  | 92   | 1995-05-12 ltclass.dtx v1.0n                                        |     |
| 1995-04-29 ltpar.dtx v1.1b                                          |      | \DeclareOption*: Use \toks@ to<br>remove need to double hash /1557  | 991 |
| General: (TO) Comments clean-up.                                    | 399  | 1995-05-12 ltfloat.dtx v1.1h                                        |     |
| 1995-05-02 ltsect.dtx v1.0l                                         |      | \@footnotemark: Add \nobreak to<br>allow hyphenation. latex/1605 .. | 932 |
| \@dottedtocline: Don't reset to<br>\rmfamily .....                  | 909  | 1995-05-12 ltpictur.dtx v1.0h                                       |     |
| 1995-05-03 ltsect.dtx v1.0m                                         |      | \picture: Macro added for<br>latex/1355 .....                       | 864 |
| General: TO: Promoted<br>documentation to doc.sty<br>standard ..... | 898  | 1995-05-12 ltvers.dtx v1.0e                                         |     |
| 1995-05-06 ltsect.dtx 1.0n                                          |      | General: Add autoload docstrip guards                               | 37  |
| \@seccntformat: Use \quad instead of<br>\hskip .....                | 904  | Check for format older than 1 year                                  | 37  |
| \@sect: Added \relax after<br>\@seccntformat just in case ..        | 902  | 1995-05-13 lfsstrc.dtx v2.3o                                        |     |
| 1995-05-07 ltboxes.dtx v1.0t                                        |      | General: Use single hash mark in<br>\DeclareOption .....            | 573 |
| General: Use \hb@xt@ .....                                          | 819  | 1995-05-16 ltfloat.dtx v1.1i                                        |     |
| 1995-05-07 ltdefns.dtx v1.2k                                        |      | \@makefnmark: Now use<br>\textrm .....                              | 929 |
| \hb@xt@: Macro added .....                                          | 81   | \textsuperscript: Command<br>added./pr1503 .....                    | 929 |
| 1995-05-07 ltmath.dtx v1.0r                                         |      | \thefootnote: Streamlined parts of<br>code. .....                   | 928 |
| General: Use \hb@xt@ .....                                          | 785  | 1995-05-17 ltboxes.dtx v1.0u                                        |     |
| 1995-05-07 ltoutput.dtx v1.1m                                       |      | \@irsbox: Removed surplus braces .                                  | 835 |
| General: Use \hb@xt@ .....                                          | 1087 | 1995-05-17 ltdefns.dtx v1.0o                                        |     |
| 1995-05-07 ltpictur.dtx v1.0g                                       |      | \g@addto@macro: Make long for<br>latex/1522 .....                   | 112 |
| General: Use \hb@xt@ .....                                          | 862  | 1995-05-17 ltlists.dtx v1.0g                                        |     |
|                                                                     |      | \@item: Removed surplus braces ...                                  | 815 |
|                                                                     |      | \@nbitem: Removed surplus braces ..                                 | 815 |
|                                                                     |      | \enumerate: Use \thr@@ and remove<br>surplus braces .....           | 817 |
|                                                                     |      | \itemize: Use \thr@@ .....                                          | 817 |

|            |                                                                                                                    |      |                                                                                                            |     |
|------------|--------------------------------------------------------------------------------------------------------------------|------|------------------------------------------------------------------------------------------------------------|-----|
| 1995-05-18 | ltfloat.dtx v1.1j                                                                                                  |      | General: Moved definition of <code>\footins</code><br>and <code>\footnoterule</code> from ltplain. . . . . | 928 |
|            | <code>\@makefnmark</code> : Added <code>\normalfont</code> . . . . .                                               | 929  |                                                                                                            |     |
|            | <code>\thempfootnote</code> : Added <code>\itshape</code> . . . . .                                                | 928  |                                                                                                            |     |
| 1995-05-19 | ltpictur.dtx v1.1a                                                                                                 |      |                                                                                                            |     |
|            | General: Support autoloading feature . . . . .                                                                     | 862  |                                                                                                            |     |
| 1995-05-20 | ltcounts.dtx v1.1b                                                                                                 |      |                                                                                                            |     |
|            | <code>\@definecounter</code> : Streamlined code . . . . .                                                          | 514  |                                                                                                            |     |
|            | <code>\@fnsymbol</code> : Allowing both text and<br>math . . . . .                                                 | 519  |                                                                                                            |     |
|            | <code>\fnsymbol</code> : Streamlined code . . . . .                                                                | 518  |                                                                                                            |     |
| 1995-05-20 | ltcounts.dtx v1.1c                                                                                                 |      |                                                                                                            |     |
|            | <code>\@definecounter</code> : And do it right . . . . .                                                           | 514  |                                                                                                            |     |
| 1995-05-20 | ltfloat.dtx v1.1k                                                                                                  |      |                                                                                                            |     |
|            | <code>\@makefnmark</code> : Moved <code>\normalfont</code><br>back and use <code>\textsuperscript</code> . . . . . | 929  |                                                                                                            |     |
|            | Moved <code>\normalfont</code> to<br><code>\textsuperscript</code> . . . . .                                       | 929  |                                                                                                            |     |
|            | <code>\textsuperscript</code> : Use <code>\normalfont</code> . . . . .                                             | 929  |                                                                                                            |     |
| 1995-05-21 | ltfssdcl.dtx v2.1t                                                                                                 |      |                                                                                                            |     |
|            | <code>\DeclareMathRadical</code> : Allow for<br>undefined cs names . . . . .                                       | 629  |                                                                                                            |     |
| 1995-05-21 | ltlists.dtx v1.0f                                                                                                  |      |                                                                                                            |     |
|            | General: Moved to doc.sty standard . . . . .                                                                       | 801  |                                                                                                            |     |
| 1995-05-21 | ltmath.dtx v1.0r                                                                                                   |      |                                                                                                            |     |
|            | <code>\sqrt</code> : Use <code>\sqrtsign</code> . . . . .                                                          | 795  |                                                                                                            |     |
|            | General: Remove <code>\mathhexbox</code> from<br>this file . . . . .                                               | 790  |                                                                                                            |     |
|            | Update some plain macros . . . . .                                                                                 | 785  |                                                                                                            |     |
|            | <code>\lefteqn</code> : Use <code>\rlap</code> . . . . .                                                           | 797  |                                                                                                            |     |
|            | <code>\righteqn</code> : Use <code>\sqrtsign</code> instead of<br><code>\sqrt</code> . . . . .                     | 787  |                                                                                                            |     |
| 1995-05-21 | ltoutenc.dtx v1.7h                                                                                                 |      |                                                                                                            |     |
|            | <code>\@inmathwarn</code> : Added several<br><code>\onlypreamble</code> . . . . .                                  | 470  |                                                                                                            |     |
| 1995-05-21 | ltoutenc.dtx v1.7j                                                                                                 |      |                                                                                                            |     |
|            | General: Updated some plain macros . . . . .                                                                       | 482  |                                                                                                            |     |
| 1995-05-21 | ltplain.dtx v1.1j                                                                                                  |      |                                                                                                            |     |
|            | General: Moved some code to other<br>files . . . . .                                                               | 14   |                                                                                                            |     |
| 1995-05-22 | ltplain.dtx v1.1k                                                                                                  |      |                                                                                                            |     |
|            | General: Definitions of <code>\footins</code> and<br><code>\footnoterule</code> moved to ltfloat. . . . .          | 33   |                                                                                                            |     |
| 1995-05-22 | lttab.dtx v1.1a                                                                                                    |      |                                                                                                            |     |
|            | General: Support autoloading feature . . . . .                                                                     | 837  |                                                                                                            |     |
| 1995-05-23 | ltfssini.dtx v2.2e                                                                                                 |      |                                                                                                            |     |
|            | <code>\newfont</code> : Font assignment made local<br>again . . . . .                                              | 656  |                                                                                                            |     |
| 1995-05-24 | ltdefns.dtx v1.1l                                                                                                  |      |                                                                                                            |     |
|            | <code>\newif</code> : (DPC) New implementation . . . . .                                                           | 87   |                                                                                                            |     |
| 1995-05-24 | ltdefns.dtx v1.2m                                                                                                  |      |                                                                                                            |     |
|            | <code>\typein</code> : (DPC) New implementation . . . . .                                                          | 82   |                                                                                                            |     |
| 1995-05-24 | ltfloat.dtx v1.1l                                                                                                  |      |                                                                                                            |     |
|            | <code>\textsuperscript</code> : Command<br>added . . . . .                                                         | 929  |                                                                                                            |     |
|            |                                                                                                                    |      | General: Moved definition of <code>\footins</code><br>and <code>\footnoterule</code> from ltplain. . . . . | 928 |
|            |                                                                                                                    |      | <code>\textsuperscript</code> : Use<br><code>\textsuperscript</code> . . . . .                             | 929 |
| 1995-05-24 | ltfssbas.dtx v3.0a                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Make file from<br>previous file, fam.dtx 1995/05/20<br>v2.2d . . . . .                              | 523  |                                                                                                            |     |
|            | <code>\mathgroup</code> : (DPC) No need to<br>redefine <code>\newfam</code> as not outer . . . . .                 | 523  |                                                                                                            |     |
| 1995-05-24 | ltfsscmp.dtx v3.0a                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Make file from<br>previous file, fam.dtx 1995/05/20<br>v2.2d . . . . .                              | 597  |                                                                                                            |     |
| 1995-05-24 | ltfssdcl.dtx v3.0a                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Make file from<br>previous file, latint.dtx 1995/05/21<br>v2.1t . . . . .                           | 602  |                                                                                                            |     |
| 1995-05-24 | ltfssini.dtx v3.0a                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Make file from<br>previous file, lfonts.dtx 1995/05/23<br>v2.2e . . . . .                           | 634  |                                                                                                            |     |
|            | <code>\cal</code> : (DPC) Remove definition . . . . .                                                              | 660  |                                                                                                            |     |
|            | <code>\mit</code> : (DPC) Remove definition . . . . .                                                              | 660  |                                                                                                            |     |
| 1995-05-24 | ltfsstrc.dtx v3.0a                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Make file from<br>previous file, tracefnt 1995/05/16<br>v2.3o . . . . .                             | 572  |                                                                                                            |     |
| 1995-05-24 | ltfsstrc.dtx v3.0b                                                                                                 |      |                                                                                                            |     |
|            | General: (DPC) Fix <code>\ProvidesFile</code><br>usage . . . . .                                                   | 572  |                                                                                                            |     |
| 1995-05-25 | ltclass.dtx v1.0p                                                                                                  |      |                                                                                                            |     |
|            | <code>\endfilecontents</code> : Delete<br><code>\filecontents</code> after preamble . . . . .                      | 1010 |                                                                                                            |     |
| 1995-05-25 | ltfilehook.dtx v1.0t                                                                                               |      |                                                                                                            |     |
|            | <code>\unquoting</code> : (CAR) added<br><code>\long</code> . . . . .                                              | 1046 |                                                                                                            |     |
| 1995-05-25 | ltfiles.dtx v1.0s                                                                                                  |      |                                                                                                            |     |
|            | <code>\document</code> : Added check for <code>\topskip</code><br>zero . . . . .                                   | 446  |                                                                                                            |     |
| 1995-05-25 | ltfiles.dtx v1.0t                                                                                                  |      |                                                                                                            |     |
|            | <code>\iffileonpath</code> : (CAR) added <code>\long</code> . . . . .                                              | 459  |                                                                                                            |     |
|            | <code>\document</code> : Corrected typo . . . . .                                                                  | 446  |                                                                                                            |     |
|            | <code>\IfFileExistsC</code> : (CAR) added <code>\long</code> . . . . .                                             | 457  |                                                                                                            |     |
|            | <code>\nofiles</code> : (CAR) added <code>\long</code> . . . . .                                                   | 449  |                                                                                                            |     |
|            | <code>\protected@write</code> : (CAR) added<br><code>\long</code> . . . . .                                        | 449  |                                                                                                            |     |
| 1995-05-25 | ltfloat.dtx v1.1m                                                                                                  |      |                                                                                                            |     |
|            | <code>\savemarbox</code> : (CAR) Reset settings<br>moved to hook . . . . .                                         | 924  |                                                                                                            |     |
|            | <code>\xfloat</code> : (CAR) Reset settings moved to<br>hook . . . . .                                             | 916  |                                                                                                            |     |

|                                                                                                                                   |                                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1995-05-25 ltlists.dtx v1.0i<br>\endtrivlist: Macros moved from<br>ltspace.dtx . . . . . 810                                      | 1995-06-09 ltoutenc.dtx v1.7i<br>\DeclareTextComposite: Rewrote<br>\DeclareTextComposite to define<br>the composite as a no-argument<br>command rather than a<br>two-argument command. . . . . 473 |
| 1995-05-25 ltmath.dtx v1.3c classes<br>\eqnnum: replace<br>\reset@font\rmfamily with<br>\normalfont (PR 1578) . . . . . 798       | 1995-06-11 ltspace.dtx v1.2g<br>\restorerelax: (CAR) \relax added to<br>stop silent eating of *. . . . . 440                                                                                       |
| 1995-05-25 ltspace.dtx v1.2f<br>\vbshack: (CAR) not used so<br>'removed'. . . . . 431                                             | 1995-06-13 ltfinal.dtx v1.0t<br>General: Add patch level string more<br>carefully . . . . . 1190                                                                                                   |
| \vspace: (CAR) \restorepar<br>added to avoid possible infinite tail<br>recursion caused by a typo in the<br>argument. . . . . 435 | Call \errorstopmode . . . . . 1192                                                                                                                                                                 |
| (CAR) macros modified to be more<br>efficient . . . . . 435                                                                       | 1995-06-13 ltpictur.dtx v1.1b<br>General: Use \ProvidesFile in<br>autoload . . . . . 862                                                                                                           |
| General: Macros moved to ltlists.dtx 421                                                                                          | 1995-06-14 ltab.dtx v1.1b<br>General: Use \ProvidesFile in<br>autoload . . . . . 837                                                                                                               |
| 1995-05-26 ltdefns.dtx v1.2n<br>\gobblefour: (CAR) Added \longs 89                                                                | 1995-06-15 lfssbas.dtx v3.0c<br>General: (DPC) minor documentation<br>changes . . . . . 523                                                                                                        |
| 1995-05-26 ltmath.dtx v1.0s<br>\eqnnum: Removed \rmfamily (PR<br>1578), replaced \reset@font with<br>\normalfont . . . . . 794    | 1995-06-15 lfsscjmp.dtx v3.0b<br>General: (DPC) minor documentation<br>edits . . . . . 597                                                                                                         |
| 1995-05-26 ltpage.dtx v1.0g<br>\psplain: removed \rmfamily (PR<br>1578) . . . . . 970                                             | 1995-06-15 lfssdcl.dtx v3.0b<br>General: (DPC) minor documentation<br>changes . . . . . 602                                                                                                        |
| 1995-05-27 lfssbas.dtx v3.0b<br>\mathgroup: (FMi) But a need to<br>define \newmathgroup . . . . . 523                             | 1995-06-19 ltbibl.dtx v1.1h<br>\bibcite: Call \newlabel so<br>repeated keys produce better<br>warning. . . . . 938                                                                                 |
| 1995-06-05 fontdef.dtx v2.2k<br>General: Moved math commands from<br>ltoutenc.dtx. . . . . 680                                    | 1995-06-19 ltclass.dtx v1.0q<br>\documentclass: Don't redefine<br>\usepackage in compat mode for<br>/1634 . . . . . 995                                                                            |
| 1995-06-05 ltfinal.dtx v1.0r<br>General: Added \MakeUppercase and<br>\MakeLowercase. . . . . 1174                                 | 1995-06-19 ltxref.dtx v1.1e<br>\newlabel: Use \newlabel to share<br>code with \bibcite . . . . . 741                                                                                               |
| 1995-06-05 ltoutenc.dtx v1.7k<br>\inmathwarn: Removed<br>\protected@cmd and replaced with<br>explicit \noexpand. . . . . 470      | 1995-06-28 lfssini.dtx v3.0b<br>General: (DPC) Fix documentation<br>typos . . . . . 634                                                                                                            |
| General: Allowed<br>\ProvideTextCommandDefault<br>after the preamble. . . . . 472                                                 | 1995-06-28 ltmath.dtx v1.0t<br>General: minor doc edits . . . . . 785                                                                                                                              |
| Commented out \textless and<br>\textgreater. . . . . 480                                                                          | 1995-07-02 lplain.dtx v1.1n<br>General: Removed surplus 'by' and '='<br>in various places . . . . . 14                                                                                             |
| Moved math commands to<br>fontdef.dtx. . . . . 482                                                                                | \offinterlineskip: Replaced 1000 by<br>\@m . . . . . 30                                                                                                                                            |
| Save some tokens in<br>\textvisiblespace and<br>\textunderscore. . . . . 480                                                      | \showoutput: Use \showoverfull to<br>save space . . . . . 33                                                                                                                                       |
| 1995-06-06 ltfinal.dtx v1.0s<br>General: Made \MakeUppercase and<br>\MakeLowercase brace their<br>argument. . . . . 1174          | \tracingall: Use \showoutput to<br>save space . . . . . 33                                                                                                                                         |

|            |                                                                                                  |    |                                                                                      |
|------------|--------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------|
| 1995-07-03 | ltdefns.dtx v1.2o                                                                                |    | Shorten redefinition of \bincite<br>and \newlabel ..... 762                          |
|            | \set@typeset@protect: Use<br>\@typeset@protect for init .....                                    | 92 |                                                                                      |
| 1995-07-03 | ltfntcmd.dtx v3.3s                                                                               |    | 1995-07-14 ltbibl.dtx v1.1i                                                          |
|            | \t@st@ic: Use clean interface for<br>jump ..... 692                                              |    | \bincite: Remove \onlypreamble so<br>still defined in new \enddocument 938           |
| 1995-07-05 | ltfntcmd.dtx v3.3s                                                                               |    | 1995-07-14 ltxref.dtx v1.1g                                                          |
|            | \t@st@ic: Renamed from \test@next 692                                                            |    | \newlabel: Remove \onlypreamble<br>so still defined in new<br>\enddocument ..... 741 |
| 1995-07-05 | ltspace.dtx v1.2h                                                                                |    | 1995-07-19 ltfssini.dtx v3.0d                                                        |
|            | \@newline: Use \break ..... 427                                                                  |    | General: (DPC) TeX2 support .... 660                                                 |
|            | \@no@pgbk: Macro replaces \pgbk and<br>\@nopgbk ..... 425                                        |    | 1995-07-20 ltboxes.dtx v1.0v                                                         |
|            | \nopagebreak: Reimplemented both<br>using \no@pgbk ..... 424                                     |    | \@isavebox: Use \sbox ..... 824                                                      |
| 1995-07-09 | ltcntrl.dtx v1.0f                                                                                |    | \@isavepicbox: Use \sbox ..... 824                                                   |
|            | \@iforloop: Reimplemented using<br>Kabelschacht method ..... 387                                 |    | 1995-07-21 ltoutput.dtx v1.1o                                                        |
|            | \@iwhiledim: Reimplemented using<br>Kabelschacht method ..... 385                                |    | \@writesetup: Command added .. 1114                                                  |
|            | \@iwhilenum: Reimplemented using<br>Kabelschacht method ..... 385                                |    | New, experimental, versions: need<br>in-lining ..... 1114                            |
|            | \@iwhilesw: Reimplemented using<br>Kabelschacht method ..... 385                                 |    | 1995-08-09 ltmath.dtx v1.0u                                                          |
|            | \@tfor: Reimplemented using<br>Kabelschacht method ..... 387                                     |    | General: Added code for class options<br>leqno and fleqn ..... 798                   |
| 1995-07-09 | ltlists.dtx v1.0j                                                                                |    | 1995-08-11 ltlenth.dtx v1.1b                                                         |
|            | enumerate: Use \expandafter .... 817                                                             |    | General: Doc typos fixed for<br>latex/753 ..... 521                                  |
|            | itemize: Use \expandafter .... 817                                                               |    |                                                                                      |
| 1995-07-12 | ltpictur.dtx v1.1d                                                                               |    | 1995-08-16 ltcntrl.dtx v1.0g                                                         |
|            | General: allow 2e commands in 209<br>mode. latex/1737 ..... 862                                  |    | \@break@tfor: Made long ..... 387                                                    |
| 1995-07-13 | ltdefns.dtx v1.0p                                                                                |    | \@forloop: Made defs long ..... 387                                                  |
|            | General: Updates to documentation .. 80                                                          |    | \@fornoop: Made defs long ..... 387                                                  |
| 1995-07-13 | ltfiles.dtx v1.0u                                                                                |    | \@iforloop: Made defs long ..... 387                                                 |
|            | General: Updates to docu ..... 442                                                               |    | \@iwhiledim: Made defs long ..... 385                                                |
| 1995-07-13 | ltfssbas.dtx v3.0d                                                                               |    | Removed \@whilenoop ..... 385                                                        |
|            | \@defaultsubs: macro added .... 543                                                              |    | \@iwhilenum: Made defs long ..... 385                                                |
|            | \@defaultsubs: macro added .... 543                                                              |    | Removed \@whilenoop ..... 385                                                        |
|            | General: minor documentation<br>changes ..... 523                                                |    | \@iwhilesw: Removed<br>\@whileswnoop ..... 385                                       |
|            | \wrong@fontshape: Change a macro<br>not a switch to flag default font<br>substitutions ..... 542 |    | \@tfor: Made defs long ..... 387                                                     |
| 1995-07-13 | ltmiscen.dtx v1.0z                                                                               |    | 1995-08-16 ltfiles.dtx v1.0v                                                         |
|            | \@centercr: Use \nobreak ..... 774                                                               |    | \document: set \maxdepth ..... 446                                                   |
|            | \@enddocument@kernel@warnings: Use<br>\@defaultsubs instead of switch 763                        |    | set \do globally ..... 446                                                           |
|            | \@writefile: Added missing percent<br>and use \relax in the THEN case 767                        |    | set \topskip globally ..... 446                                                      |
|            | \@xobeysp: Use \nobreak ..... 777                                                                |    | 1995-08-24 ltfssbas.dtx v3.0f                                                        |
|            | General: Improve Documentation .. 760                                                            |    | General: Added autoload code .... 523                                                |
|            | \enddocument: Set \setckpt to<br>\gobble@font@spec instead of defining it<br>by hand ..... 761   |    | 1995-08-24 ltfsstrc.dtx v3.0c                                                        |
|            |                                                                                                  |    | General: Macro \gobble@font@spec<br>removed ..... 586                                |
|            |                                                                                                  |    | \tryis@simple: ..... 593                                                             |
|            |                                                                                                  |    | 1995-08-25 ltoutput.dtx v1.1p                                                        |
|            |                                                                                                  |    | General: Support autoloading feature<br>(FMi). ..... 1087                            |
|            |                                                                                                  |    | 1995-09-01 lterror.dtx v1.2i                                                         |
|            |                                                                                                  |    | General: Add autoload support ... 388                                                |
|            |                                                                                                  |    | 1995-09-01 lplain.dtx v1.1m                                                          |
|            |                                                                                                  |    | \empty: Use \let to save space .... 27                                               |
|            |                                                                                                  |    | \I: Use \let to save space ..... 27                                                  |

|            |                                                                                      |     |            |                                                                                                    |                   |
|------------|--------------------------------------------------------------------------------------|-----|------------|----------------------------------------------------------------------------------------------------|-------------------|
| 1995-09-14 | ltplain.dtx v1.1o                                                                    |     | 1995-10-11 | ltoutput.dtx v1.1r                                                                                 |                   |
|            | General: Moved \multispan to<br>ltab.dtx . . . . .                                   | 14  |            | \clearpage: Added a check so that it<br>does not lose the argument of<br>\twocolumn[...] . . . . . |                   |
| 1995-09-14 | lttab.dtx v1.1c                                                                      |     | 1995-10-16 | ltbibl.dtx v1.1j                                                                                   |                   |
|            | \cline: (DPC) New implementation                                                     | 860 |            | \cite: (DPC) Make robust . . . . .                                                                 |                   |
| 1995-09-15 | ltfssini.dtx v3.0e                                                                   |     | 1995-10-16 | ltboxes.dtx v1.0w                                                                                  |                   |
|            | General: (DPC) Modify TeX2<br>message . . . . .                                      | 660 |            | General: Clarify makebox description                                                               |                   |
| 1995-09-19 | ltmiscen.dtx v1.1a                                                                   |     | 1995-10-16 | ltdefns.dtx v1.2u                                                                                  |                   |
|            | \verb: Put \noligs after<br>\verbatim@font where it belongs.                         | 783 |            | \@ifstar: (DPC) New<br>implementation, for /1910 . . . . .                                         |                   |
| 1995-10-01 | ltfiles.dtx LaTeXe                                                                   |     | 1995-10-16 | \new@command: (DPC) Use \@testopt<br>/1911 . . . . .                                               |                   |
|            | \@addtofilelist: Macro added . . .                                                   | 463 |            | \new@environment: (DPC) Use<br>\@testopt /1911 . . . . .                                           |                   |
| 1995-10-02 | ltdefns.dtx v1.2q                                                                    |     | 1995-10-16 | \typein: (DPC) Use \@testopt /1911 . . .                                                           |                   |
|            | \@ifnch: Use \@let@token for<br>internal/924, save \reserved@e .                     | 108 |            | \reset@font: Added \relax after<br>\usefont, as the latter eats up<br>spaces. . . . .              |                   |
|            | \@ifnextchar: Use \@let@token . . .                                                  | 107 | 1995-10-16 | \yeqncr: (DPC) Use \@testopt<br>/1911 . . . . .                                                    |                   |
|            | \@protected@testopt: Macro added .                                                   | 85  |            | \sqrt: (DPC) Make robust /1808 . .                                                                 |                   |
|            | \@testopt: Macro added . . . . .                                                     | 84  | 1995-10-16 | \newlinebreak: (DPC) Use \@testopt<br>/1911 . . . . .                                              |                   |
|            | \@xargdef: New implementation,<br>using \@test@opt . . . . .                         | 84  |            | \nopagebreak: (DPC) Use \@testopt<br>/1911 . . . . .                                               |                   |
| 1995-10-03 | fontdef.dtx v2.2l                                                                    |     | 1995-10-16 | \thinspace: (DPC) Revert to previous<br>\newtheorem behaviour . . . . .                            |                   |
|            | General: @@sqrt from patch file for<br>/1701 . . . . .                               | 662 |            | 1995-10-17                                                                                         | ltclass.dtx v1.0r |
| 1995-10-03 | ltdefns.dtx v1.2r                                                                    |     |            | \@providesfile: Delay definition of<br>\ProvidesFile till ltfinal . . . . .                        |                   |
|            | \typein: Add missing \@typein for<br>/1710 (from patch file) . . . . .               | 82  |            | \ProcessOptions*: Reset<br>\CurrentOption for<br>graphics/1873 . . . . .                           |                   |
| 1995-10-03 | ltpictur.dtx v1.1e                                                                   |     | 1995-10-17 | ltdirchk.dtx v1.0l                                                                                 |                   |
|            | General: New autoload code . . . . .                                                 | 862 |            | General: Modify initex version of<br>\ProvidesFile . . . . .                                       |                   |
| 1995-10-04 | ltfssbas.dtx v3.0g                                                                   |     | 1995-10-17 | ltfinal.dtx v1.0v                                                                                  |                   |
|            | General: Modify autoload code . . . .                                                | 523 |            | \@providesfile: reset macro . . . . .                                                              |                   |
| 1995-10-04 | ltfsstrc.dtx v3.0d                                                                   |     |            | \reserved@b: reset here after the<br>\input above . . . . .                                        |                   |
|            | General: (DPC) Modify autoload<br>code . . . . .                                     | 572 | 1995-10-17 | ltplain.dtx v1.1s                                                                                  |                   |
| 1995-10-04 | lttab.dtx v1.1d                                                                      |     |            | \reject: Move \supereject to compat<br>file . . . . .                                              |                   |
|            | General: Modify autoload support .                                                   | 837 | 1995-10-17 | lttab.dtx v1.1e                                                                                    |                   |
| 1995-10-06 | ltfiles.dtx v1.0w                                                                    |     |            | \@cline: (DPC) Use \@multicnt . . . . .                                                            |                   |
|            | \@missingfileerror: Autoload error                                                   | 461 |            | \@multispan: (DPC) Macro added.                                                                    |                   |
| 1995-10-09 | lterror.dtx v1.2j                                                                    |     | 1995-10-19 | ltfinal.dtx v1.0w                                                                                  |                   |
|            | General: Modify autoload support .                                                   | 388 |            | \@filelist: Move after \reserved@a<br>setting . . . . .                                            |                   |
| 1995-10-09 | ltoutenc.dtx v1.7m                                                                   |     |            | 1192                                                                                               |                   |
|            | \@inmathwarn: Autoload error . . .                                                   | 471 |            |                                                                                                    |                   |
| 1995-10-10 | ltfssbas.dtx v3.0h                                                                   |     |            |                                                                                                    |                   |
|            | \showhyphens: Use \normalfont and<br>make colour safe, and<br>autoloadable . . . . . | 547 |            |                                                                                                    |                   |
| 1995-10-10 | ltfssdcl.dtx v3.0c                                                                   |     |            |                                                                                                    |                   |
|            | \non@alpherr: (DPC) autoload error<br>message . . . . .                              | 606 |            |                                                                                                    |                   |
| 1995-10-10 | ltplain.dtx v1.1r                                                                    |     |            |                                                                                                    |                   |
|            | General: Autoload tracing code . . .                                                 | 14  |            |                                                                                                    |                   |
| 1995-10-10 | ltthm.dtx v1.0f                                                                      |     |            |                                                                                                    |                   |
|            | General: Make \newtheorem 'only<br>preamble' . . . . .                               | 894 |            |                                                                                                    |                   |

|            |                                                                                                                     |      |                                                                                                                 |                    |
|------------|---------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------|--------------------|
| 1995-10-20 | ltbibl.dtx v1.1k                                                                                                    |      | \@setref: Switch for refundefined<br>renamed . . . . .                                                          | 740                |
|            | \@citex: Removed refundefined flag                                                                                  | 939  |                                                                                                                 |                    |
|            | \nocite: Removed refundefined flag                                                                                  | 940  | \if@multiplelabels: Macro removed                                                                               | 742                |
| 1995-10-20 | ltclass.dtx v1.0s                                                                                                   |      |                                                                                                                 |                    |
|            | \@begindocumenthook: Make setting<br>conditional, for autoload version                                              | 1007 | 1995-10-25                                                                                                      | ltalloc.dtx v1.1b  |
| 1995-10-20 | ltfssbas.dtx v3.0i                                                                                                  |      | General: General doc improvements                                                                               | 382                |
|            | General: (DPC) Modify autoload code,<br>change \undefined . . . . .                                                 | 523  |                                                                                                                 |                    |
| 1995-10-20 | ltfsstrc.dtx v3.0e                                                                                                  |      | \@endfloatbox: (CAR) macro added:<br>to unify code for double and single<br>versions . . . . .                  | 920                |
|            | General: (DPC) Modify autoload<br>code . . . . .                                                                    | 572  | \end@dblfloa: (CAR) unify code for<br>double and single versions . . . . .                                      | 919                |
| 1995-10-22 | ltfssbas.dtx v3.0j                                                                                                  |      | \end@float: (CAR) unify code for<br>double and single versions . . . . .                                        | 918                |
|            | General: (RmS) New size function<br>macro \genb@sfcnt needs to be<br>disabled at \document. . . . .                 | 523  | 1995-10-25                                                                                                      | ltidxglo.dtx v1.1d |
| 1995-10-22 | ltfsstrc.dtx v3.0f                                                                                                  |      | General: Doc cleanup . . . . .                                                                                  | 934                |
|            | General: Added 'genb' and 'sgenb' size<br>functions to support new DC font<br>naming scheme. . . . .                | 572  | 1995-10-25                                                                                                      | ltsect.dtx v1.0q   |
| 1995-10-23 | lttab.dtx v1.1f                                                                                                     |      | \subparagraphmark: Use \let not<br>\def to save space. . . . .                                                  | 906                |
|            | \@settab: (CAR) Ensure that<br>\hightab increases by at most<br>one . . . . .                                       | 844  | 1995-10-27                                                                                                      | lpictur.dtx v1.1f  |
|            | \@startline: (CAR) Ensure that<br>\nxttabmar is never larger than<br>\hightab . . . . .                             | 842  | General: Move initialization to kernel<br>from autoload file . . . . .                                          | 889                |
|            | \poptabs: (CAR) Ensure that<br>\curtab is never larger than<br>\hightab . . . . .                                   | 845  | 1995-10-31                                                                                                      | ltboxes.dtx v1.0x  |
|            | \tabbing: (CAR) Make \hightab<br>consistently a local variable . . .                                                | 844  | \@finalstrut: Add \nobreak in horiz<br>mode to allow hyphenation.<br>internal/1931 . . . . .                    | 835                |
| 1995-10-24 | ltfiles.dtx v1.1a                                                                                                   |      | 1995-11-01                                                                                                      | fontdef.dtx v2.2m  |
|            | \document: Removed multiplelabels<br>switch . . . . .                                                               | 445  | General: add \nfss@catcodes for<br>internal/1932 . . . . .                                                      | 665                |
|            | Removed refundefined switch . . .                                                                                   | 445  | 1995-11-01                                                                                                      | ltdirchk.dtx v1.0n |
| 1995-10-24 | ltfssbas.dtx v3.0k                                                                                                  |      | General: Initialise \@addtofilelist to<br>\gobble . . . . .                                                     | 4                  |
|            | \@defaultsubs: macro removed . .                                                                                    | 543  | 1995-11-01                                                                                                      | ltfinal.dtx v1.0x  |
|            | \wrong@fontshape: Make this code<br>inline since it happens only here                                               | 542  | General: (DPC) Switch meaning of<br>\@addtofilelist for cfg files . .                                           | 1180               |
| 1995-10-24 | ltmisen.dtx v1.1b                                                                                                   |      | 1995-11-01                                                                                                      | ltfssbas.dtx v3.0m |
|            | \@enddocument@kernel@warnings:<br>Changed logic for producing<br>warning messages and removed<br>switch . . . . .   | 763  | \DeclareFontShape@: (DPC) Test for<br>\relax not \undefined,<br>internal/1933 . . . . .                         | 524                |
|            | Use \@refundefined instead of<br>switch . . . . .                                                                   | 763  | 1995-11-01                                                                                                      | ltfssini.dtx v3.0g |
| 1995-10-24 | ltxref.dtx v1.1h                                                                                                    |      | General: (DPC) Switch meaning of<br>\@addtofilelist for cfg files . .                                           | 660                |
|            | \@multiplelabels: Switch for<br>multiplelabels removed . . . . .                                                    | 742  | 1995-11-02                                                                                                      | ltfssbas.dtx v3.0n |
|            | \@newl@bel: Switch for multiplelabels<br>replaced by inline code . . . . .                                          | 741  | \wrong@fontshape: (DPC) Remove<br>extra space with \string for<br>latex/1676 . . . . .                          | 542                |
|            | \@refundefined: Switch for<br>refundefined replaced . . . . .                                                       | 739  | 1995-11-02                                                                                                      | ltoutenc.dtx v1.7n |
|            |                                                                                                                     |      | General: Changed internal name \a to<br>\@tabacckludge to protect against<br>redefinition by malicious users. . | 478                |
| 1995-11-07 | ltlists.dtx v1.0k                                                                                                   |      |                                                                                                                 |                    |
|            | \@doendpe: Enclosed \setbox0<br>assignment by a group so that it<br>leaves the contents of box 0<br>intact. . . . . | 812  |                                                                                                                 |                    |

|                               |                                                                                                                                            |      |                               |                                                                                                                                                                                                 |          |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1995-11-07 ltoutenc.dtx v1.7o | General: Added <code>\leavevmode</code> at start of <code>\c</code> , otherwise the output routine might be invoked within the macro. .... | 482  | 1995-11-29 ltoutenc.dtx v1.7t | General: Added <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textbackslash</code> , <code>\textbar</code> , <code>\textgreater</code> and <code>\textless</code> . .... | 486      |
|                               | Changed <code>\char32</code> to <code>\@xxxii</code> (two tokens less). ....                                                               | 483  |                               | Added <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textregistered</code> and <code>\texttrademark</code> . ....                                                        | 480      |
|                               | Replaced octal number 27 by decimal number 23 to protect against the quote character being active. ....                                    | 483  |                               | Added <code>\textbackslash</code> and <code>\textbar</code> . ....                                                                                                                              | 479, 490 |
|                               | Replaced some 0's by <code>\z@</code> (faster). ....                                                                                       | 483  |                               | Added <code>\textless</code> and <code>\textgreater</code> . ....                                                                                                                               | 480, 491 |
| 1995-11-10 ltoutput.dtx v1.1s | <code>\@shipoutsetup</code> : Command removed ....                                                                                         | 1114 | 1995-12-01 ltoutenc.dtx v1.7u | General: Made <code>\\$S</code> a Default, rather than having the default point to the OT1 definition. ....                                                                                     | 480      |
|                               | <code>\@writeshop</code> : Command removed In-lined ....                                                                                   | 1114 | 1995-12-04 ltspace.dtx v1.2k  | <code>\nobreakspace</code> : Macro added ....                                                                                                                                                   | 437      |
| 1995-11-14 ltclass.dtx v1.0t  | <code>\@unprocessedoptions</code> : Allow empty option ....                                                                                | 1009 | 1995-12-04 ltspace.dtx v1.2l  | <code>\@obeysp</code> : braces added to definition of tilde ....                                                                                                                                | 438      |
|                               | <code>\@loadwithoptions</code> : macro added ....                                                                                          | 996  | 1995-12-04 preload.dtx v2.4e  | General: Ulrik Vieth. added 12pt OMS and OML preloads /1989 .                                                                                                                                   | 685      |
|                               | <code>\LoadClassWithOptions</code> : macro added ....                                                                                      | 996  | 1995-12-05 ltdefns.dtx 1.2w   | <code>\@unexpandable@protect</code> : Removed <code>\unexpandable@noexpand</code> as never used. internal/1733 ....                                                                             | 90       |
|                               | <code>\RequirePackageWithOptions</code> : macro added ....                                                                                 | 997  | 1995-12-05 ltfiles.dtx v1.1c  | <code>\document</code> : <code>\ignorespaces</code> added for latex/1933 ....                                                                                                                   | 446      |
| 1995-11-17 ltfssbas.dtx v3.0m | <code>\@wrong@font@char</code> : (DPC) Macro added. latex/1676 ....                                                                        | 543  | 1995-12-05 ltfloat.dtx v1.1n  | <code>\@textsuperscript</code> : Use <code>\ensuremath</code> for latex/1984. ....                                                                                                              | 929      |
|                               | <code>\define@newfont</code> : Redefine <code>\typeout</code> latex/1676 ....                                                              | 537  | 1995-12-05 ltoutenc.dtx v1.7v | <code>\@inmathwarn</code> : Changed <code>\TextSymbolUnavailable</code> text .                                                                                                                  | 471      |
|                               | <code>\@wrong@fontshape</code> : Support <code>\@wrong@font@char</code> latex/1676 .                                                       | 542  | 1995-12-06 ltfssbas.dtx v3.00 | <code>\nfss@catcodes</code> : Reset hat, for typeouts etc in fd files ....                                                                                                                      | 538      |
| 1995-11-17 ltoutenc.dtx v1.7p | <code>\UseTextSymbol</code> : Support <code>\@wrong@font@char</code> latex/1676 .                                                          | 475  | 1995-12-07 ltbibl.dtx v1.1l   | <code>\@citex</code> : Restored name of <code>\G@refundefinedtrue</code> ....                                                                                                                   | 939      |
| 1995-11-18 ltoutenc.dtx v1.7q | <code>\UseTextSymbol</code> : Modify message slightly ....                                                                                 | 475  | 1995-12-07 ltfloat.dtx v1.1m  | <code>\@textsuperscript</code> : Move <code>\m@th</code> out of the <code>\ensuremath</code> for latex/1984. ....                                                                               | 929      |
| 1995-11-21 fontdef.dtx v2.2n  | General: Incorporate changed figures, as in plain.tex ....                                                                                 | 678  | 1995-12-07 ltref.dtx v1.1i    | <code>\@setref</code> : Switch for refundefined restored ....                                                                                                                                   | 740      |
| 1995-11-27 ltfssbas.dtx v3.0n | <code>\nfss@catcodes</code> : Reset hash, for definitions in fd files ....                                                                 | 538  |                               | <code>\G@refundefinedtrue</code> : Renamed (back) from <code>\G@refundefined</code> ..                                                                                                          | 739      |
| 1995-11-28 ltfloat.dtx v1.1n  | General: documentation fixes ....                                                                                                          | 911  |                               |                                                                                                                                                                                                 |          |
| 1995-11-28 lfsstrc.dtx v3.0g  | General: documentation fixes ....                                                                                                          | 572  |                               |                                                                                                                                                                                                 |          |
| 1995-11-28 ltoutenc.dtx v1.7r | General: Added math mode checks to text commands. ....                                                                                     | 470  |                               |                                                                                                                                                                                                 |          |
|                               | doc fixes ....                                                                                                                             | 466  |                               |                                                                                                                                                                                                 |          |
|                               | Renamed <code>\@changed@x@err</code> to <code>\TextSymbolUnavailable</code> . ....                                                         | 470  |                               |                                                                                                                                                                                                 |          |

|                                                                                                          |          |                                                                                                                                                                   |      |
|----------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 1995-12-11 ltoutenc.dtx v1.7w                                                                            |          | 1996-05-21 ltoutenc.dtx v1.7y                                                                                                                                     |      |
| General: Modified \copyright . . . . .                                                                   | 480      | General: Corrected error message<br>(CAR) . . . . .                                                                                                               | 509  |
| 1995-12-13 ltdefns.dtx 1.2x                                                                              |          | 1996-05-21 ltsect.dtx v1.0s                                                                                                                                       |      |
| \:-: Documentation changed. . . . .                                                                      | 110      | \@sect: (DPC) Added extra braces for<br>internal/2148 . . . . .                                                                                                   | 902  |
| 1996-01-10 ltfiles.dtx v1.1d                                                                             |          | (DPC) Moved brace to allow<br>commands like \MakeUppercase in<br>6th argument. Changed \par to<br>\endgraf to allow non-long<br>commands. internal/2148 . . . . . | 902  |
| \@iffileonpath: Change argument<br>handling to not require doubled<br>hash. latex/2024 . . . . .         | 459      | \@ssect: (DPC) Added extra braces<br>for internal/2148 . . . . .                                                                                                  | 905  |
| 1996-01-20 ltidxglo.dtx v1.1e                                                                            |          | (DPC) Moved brace to allow<br>commands like \MakeUppercase in<br>4th argument. Changed \par to<br>\endgraf to allow non-long<br>commands. internal/2148 . . . . . | 905  |
| \makeglossary: Make no-op after use<br>pr/2048 . . . . .                                                 | 935      | 1996-05-23 ltoutenc.dtx v1.7z                                                                                                                                     |      |
| \makeindex: Make no-op after use<br>pr/2048 . . . . .                                                    | 935      | \@strip@args: \expandafter added to<br>match other changes for<br>latex/2133 . . . . .                                                                            | 475  |
| 1996-01-20 ltspace.dtx v1.2m                                                                             |          | \add@accent: macro added.<br>latex/2133 . . . . .                                                                                                                 | 472  |
| \vspace: Made robust . . . . .                                                                           | 435      | \DeclareTextAccent: Reimplemented<br>using \add@accent to save space<br>latex/2133 . . . . .                                                                      | 472  |
| 1996-03-25 ltmath.dtx v1.1a                                                                              |          | \DeclareTextCompositeCommand:<br>Modified to cope with new<br>\add@accent command: required<br>removal of check for one<br>argument-command . . . . .             | 473  |
| \@ensuredmath: Macro added for<br>amslatex/2104 . . . . .                                                | 797      | 1996-05-24 ltoutput.dtx v1.1t                                                                                                                                     |      |
| \ensuremath: Reimplement for<br>amslatex/2104 . . . . .                                                  | 797      | \@specialoutput: Check that<br>\@colroom is less than \vsize,<br>indicating that a float has been<br>added . . . . .                                              | 1105 |
| 1996-04-18 ltpage.dtx v1.0i                                                                              |          | Cut-off point changed to<br>1.5\baselineskip . . . . .                                                                                                            | 1105 |
| General: Improve documentation . . . . .                                                                 | 969      | \@topnewpage: Cut-off point changed<br>to 2.5\baselineskip . . . . .                                                                                              | 1104 |
| 1996-04-22 ltmiscen.dtx v1.1c                                                                            |          | 1996-05-25 ltoutput.dtx v1.1u                                                                                                                                     |      |
| General: Improve Documentation . . . . .                                                                 | 760      | \@specialoutput: Correct the above<br>check . . . . .                                                                                                             | 1105 |
| 1996-04-22 ltspace.dtx v1.2n                                                                             |          | 1996-06-03 ltmiscen.dtx v1.1d                                                                                                                                     |      |
| General: Documentation<br>Improvements . . . . .                                                         | 421      | \overline: Exchanged the following<br>two code lines so that \dospecials<br>cannot reset the category code of<br>characters handled by \noligs. . . . .           | 778  |
| 1996-04-22 lttab.dtx v1.1g                                                                               |          | General: Move setting of verbatim font<br>and \noligs. . . . .                                                                                                    | 760  |
| \@tabclassz: (DPC) Extra \hskip<br>keeps tabcolsep in empty columns<br>internal/2122 . . . . .           | 857      | \verb: Put setting of verbatim font<br>after \dospecials so that<br>\dospecials cannot reset the                                                                  |      |
| 1996-04-23 ltcnts.dtx v1.1d                                                                              |          |                                                                                                                                                                   |      |
| General: Documentation<br>improvements . . . . .                                                         | 512      |                                                                                                                                                                   |      |
| 1996-04-24 ltfiles.dtx v1.1e                                                                             |          |                                                                                                                                                                   |      |
| \document: (DPC) Reset<br>\AtBeginDocument eg for<br>latex/1297 . . . . .                                | 445      |                                                                                                                                                                   |      |
| 1996-05-08 lfsstrc.dtx v3.0h                                                                             |          |                                                                                                                                                                   |      |
| \math@egroup: Use \bgroup instead of<br>\begingroup to match a kernel<br>change made in 1994!! . . . . . | 584      |                                                                                                                                                                   |      |
| 1996-05-09 lftntcmd.dtx v3.3t                                                                            |          |                                                                                                                                                                   |      |
| \check@icr: Default definitions<br>added . . . . .                                                       | 690      |                                                                                                                                                                   |      |
| 1996-05-17 fontdef.dtx v2.2o                                                                             |          |                                                                                                                                                                   |      |
| General: \@sqrt removed, at<br>last . . . . .                                                            | 662, 678 |                                                                                                                                                                   |      |
| 1996-05-17 ltfiles.dtx v1.1f                                                                             |          |                                                                                                                                                                   |      |
| \nofiles: added \write to<br>\protected@write for latex/2146                                             | 449      |                                                                                                                                                                   |      |
| 1996-05-18 ltoutenc.dtx v1.7x                                                                            |          |                                                                                                                                                                   |      |
| General: Produce error if encoding not<br>found. pr/2054 . . . . .                                       | 509      |                                                                                                                                                                   |      |

|                                                                                                        |      |                                                                                                                                               |      |
|--------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------|------|
| category code of characters<br>handled by \noligs. ....                                                | 783  | \nfss@catcodes: omit \relax as not<br>needed ....                                                                                             | 538  |
| 1996-06-10 ltboxes.dtx v1.0y<br>\parboxto: (DPC) Changed<br>\endgraf to \par ....                      | 828  | 1996-07-26 ltfssdcl.dtx v3.0e<br>\init@restore@version: Removed<br>\ifrestore@version switch and<br>replaced by<br>\init@restore@version .... | 606  |
| 1996-06-10 ltsect.dtx v1.0t<br>\sect: (DPC) Changed \endgraf to<br>\par ....                           | 902  | 1996-07-26 lfsstrc.dtx v3.0i<br>\init@restore@glb@settings: macro<br>added replacing \if@inmath<br>switch ....                                | 583  |
| 1996-06-13 ltdirchk.dtx v1.0r<br>General: documentation improvements<br>mainly from internal/2174 .... | 1    | 1996-07-26 ltlsts.dtx v1.0l<br>\item: Remove unnecessary \global<br>before \minipage... ....                                                  | 814  |
| 1996-06-14 lttab.dtx v1.1h<br>\tabclassz: (DPC) Change<br>both\z@skip to 1sp for<br>latex/2160 ....    | 857  | Remove unnecessary \global before<br>\nobreak....                                                                                             | 815  |
| 1996-06-22 ltspace.dtx v1.2o<br>General: Documentation of problems<br>added ....                       | 421  | 1996-07-26 ltmath.dtx v1.1b<br>General: Removed \global before<br>\ignorestrue in various places.                                             | 785  |
| 1996-07-10 ltfinal.dtx v1.0y<br>\toks: Free up memory from scratch<br>registers /2213 ....             | 1191 | \ignorefalse: put \global into<br>definition ....                                                                                             | 761  |
| 1996-07-19 ltoutenc.dtx v1.8a<br>\strip@args: Use char 0 not @ as<br>carrier for \lowercase /2197 ...  | 475  | \begin: remove \global before<br>\ignore... ....                                                                                              | 769  |
| 1996-07-26 ltboxes.dtx v1.0z<br>\if@minipage: put \global into<br>definition ....                      | 830  | \end: remove \global before<br>\ignore... ....                                                                                                | 771  |
| 1996-07-26 ltclass.dtx v1.0u<br>\classoptionslist: made only<br>preamble ....                          | 979  | \ignorespacesafterend: user level<br>macro added ....                                                                                         | 761  |
| 1996-07-26 ltdefns.dtx v1.2y<br>\reargdef: third arg picked up by<br>\yargdef ....                     | 85   | 1996-07-26 ltoutput.dtx v1.1v<br>\testfp: remove \global before<br>\test... ....                                                              | 1149 |
| \renew@command: use \noexpand<br>instead of \string ....                                               | 86   | \xtryfc: remove \global before<br>\test... ....                                                                                               | 1123 |
| use \relax in place of empty arg ..                                                                    | 86   | \ztryfc: remove \global before<br>\test... ....                                                                                               | 1124 |
| \renew@environment: use \relax in<br>place of empty arg ....                                           | 87   | General: put \global into definition<br>remove \global before \test... ....                                                                   | 1097 |
| 1996-07-26 ltfloat.dtx v1.1n<br>\endfloatbox: remove unnecessary<br>\global before \minipage... .      | 920  | \clearpage: add number of missing<br>percents ....                                                                                            | 1100 |
| \savemarbox: remove unnecessary<br>\global before \minipage... .                                       | 924  | 1996-07-26 ltplain.dtx v1.1t<br>\sh@ft: replace \dimen\z@ by \dimen@                                                                          | 32   |
| \setminipage: remove unnecessary<br>\global before \minipage... .                                      | 918  | 1996-07-26 ltsect.dtx v1.0u<br>\starttoc: removed \global before<br>\nobreak....                                                              | 907  |
| \setnobreak: remove unnecessary<br>\global before \nobreak... .                                        | 918  | \xsect: Removed \global before<br>\nobreak....                                                                                                | 903  |
| 1996-07-26 ltssbas.dtx v3.0p<br>\DeclareMathSizes: use faster \if<br>test ....                         | 530  | 1996-07-26 ltspace.dtx v1.2p<br>\if@nobreak: put \global inside<br>definition ....                                                            | 427  |
|                                                                                                        |      | 1996-07-27 ltssbas.dtx v3.0q<br>General: \if@inmath switch removed                                                                            | 536  |
|                                                                                                        |      | 1996-07-27 ltspace.dtx v1.2q<br>General: Further documentation of<br>problems ....                                                            | 421  |

|                                                                                                                           |      |                                                                                                                                                                                     |            |
|---------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1996-07-27 ltspace.dtx v1.2r                                                                                              |      | 1996-10-21 ltab.dtx v1.1i                                                                                                                                                           |            |
| General: Correct documentation of<br>problems . . . . .                                                                   | 421  | \array: Use \set@typeset@protect . . . . .                                                                                                                                          | 850        |
| 1996-08-02 ltfloat.dtx v1.1o                                                                                              |      | General: Moved the code associated<br>with \cmkpream into the group<br>provided by the box, for<br>robustness (latex/2183) . . . . .                                                | 849        |
| \@xmpar: Remove \global before<br>\@ignore . . . . .                                                                      | 925  | \multicolumn: Make \multicolumn<br>long (latex/2180) . . . . .                                                                                                                      | 852        |
| 1996-08-02 ltsect.dtx v1.0v                                                                                               |      | \tabbing: Moved the \indent so that<br>the \everypar can remove it when<br>necessary; this is needed because<br>the code for items in lists has<br>changed (see pr/22111) . . . . . | 844        |
| \@afterheading: Removed \global<br>before \nobreak . . . . .                                                              | 905  | 1996-10-23 llists.dtx v1.0m                                                                                                                                                         |            |
| 1996-08-02 ltspace.dtx v1.2s                                                                                              |      | \item: \nobreak... moved into the<br>\everypar and not executed<br>unconditionally, see above . . . . .                                                                             | 815        |
| \@EspHack: Remove \global before<br>\@ignore . . . . .                                                                    | 430  | \kern... changed to \setbox... . . . . .                                                                                                                                            | 814        |
| 1996-08-25 lfssbas.dtx v3.0r                                                                                              |      | Added setting of \clubpenalty and<br>set \nobreakfalse only when<br>necessary . . . . .                                                                                             | 815        |
| \nfss@catcodes: Reset the acute,<br>grave and double quote chars as<br>well . . . . .                                     | 538  | 1996-10-23 ltdirchk.dtx v1.0t                                                                                                                                                       |            |
| 1996-09-21 ltoutput.dtx v1.1w                                                                                             |      | \@xsect: Replaced \hskip... with<br>\setbox... as used in<br>\@afterheading . . . . .                                                                                               | 903        |
| \@writesetup: Added<br>\@parboxrestore and made<br>consequent deletions: wait for the<br>howls of protest . . . . .       | 1114 | 1996-10-24 ltboxes.dtx v1.1a                                                                                                                                                        |            |
| 1996-09-25 ltdirchk.dtx v1.0t                                                                                             |      | \array@parboxrestore: Added local<br>settings of flags: dangerous! . . . . .                                                                                                        | 829        |
| General: Move ltxcheck to separate file                                                                                   | 13   | \@iiiminipage: Use it or lose it<br>(@setminpage): Frank will want to<br>lose it . . . . .                                                                                          | 831        |
| 1996-09-28 ltmiscen.dtx v1.1f                                                                                             |      | 1996-10-24 ltfloat.dtx v1.1p                                                                                                                                                        |            |
| \@xobeysp: Moved to ltspace.dtx . . .                                                                                     | 777  | \@floatboxreset: Added local<br>settings of flags: dangerous! . . . . .                                                                                                             | 918        |
| 1996-09-28 ltspace.dtx v1.2t                                                                                              |      | \@marginparreset: Added local<br>settings of flags: dangerous! . . . . .                                                                                                            | 924        |
| \@xobeysp: Moved from ltmiscen.dtx<br>and redefined to use<br>\nobreakspace . . . . .                                     | 437  | \@xfloat: Added \nодокумент to<br>trap floats in the preamble . . . . .                                                                                                             | 915        |
| 1996-09-29 ltfiles.dtx v1.1g                                                                                              |      | 1996-10-24 ltoutput.dtx v1.1z                                                                                                                                                       |            |
| \document: Added disabling of<br>\nодокумент . . . . .                                                                    | 446  | \@addtocurcol: Added \nobreak, etc<br>as appropriate . . . . .                                                                                                                      | 1129, 1132 |
| 1996-09-29 ltoutput.dtx v1.1x                                                                                             |      | \@specialoutput: Added \nobreak as<br>appropriate . . . . .                                                                                                                         | 1107       |
| \newpage: Checks for noskipsec and<br>inlabel added . . . . .                                                             | 1101 | \@topnewpage: Added \nодокумент<br>to trap \twocolumn in the<br>preamble . . . . .                                                                                                  | 1103       |
| 1996-09-29 ltsect.dtx 1.0w                                                                                                |      | \newpage: Better checks for noskipsec<br>and inlabel added, plus nobreak . . . . .                                                                                                  | 1101       |
| \@noskipsetrue: Added<br>documentation . . . . .                                                                          | 899  | 1996-10-25 llists.dtx v1.0n                                                                                                                                                         |            |
| 1996-09-30 ltoutput.dtx v1.1y                                                                                             |      | \endtrivlist: Change \indent to<br>\leavevmode . . . . .                                                                                                                            | 810        |
| \newpage: Checks for noskipsec and<br>inlabel removed pending further<br>tests . . . . .                                  | 1101 | Reset flags explicitly . . . . .                                                                                                                                                    | 810        |
| 1996-10-04 ltclass.dtx v1.0v                                                                                              |      | 1996-10-25 ltoutput.dtx v1.2a                                                                                                                                                       |            |
| \RequirePackageWithOptions: Reset<br>\@unprocessedoptions for /2269                                                       | 997  | \newpage: Reset all flags explicitly . . . . .                                                                                                                                      | 1101       |
| 1996-10-05 ltfiles.dtx v1.1h                                                                                              |      |                                                                                                                                                                                     |            |
| \clubpenalty: Added setting its<br>value . . . . .                                                                        | 444  |                                                                                                                                                                                     |            |
| 1996-10-08 lfntcmd.dtx v3.3u                                                                                              |      |                                                                                                                                                                                     |            |
| \DeclareTextFontCommand: Removed<br>\check@icr when in vmode since<br>it causes various errors (see<br>pr/2157) . . . . . | 688  |                                                                                                                                                                                     |            |

|                                                                                                           |               |                                                                                              |     |
|-----------------------------------------------------------------------------------------------------------|---------------|----------------------------------------------------------------------------------------------|-----|
| 1996-10-26 ltlists.dtx v1.0o                                                                              |               | 1996-11-18 ltoutenc.dtx v1.8d                                                                |     |
| \endtrivlist: Correct typo . . . . .                                                                      | 810           | General: (DPC) lowercase external file<br>names. internal/1044 . . . . .                     | 509 |
| 1996-10-27 ltoutenc.dtx v1.8c                                                                             |               | 1996-11-20 fontdef.dtx v2.2p                                                                 |     |
| \@strip@args: Removed macro . . . . .                                                                     | 473           | General: lowercase fd and enc.def file<br>names /1044 . . . . .                              | 662 |
| General: Added \r A . . . . .                                                                             | 483           | 1996-11-20 ltvers.dtx v1.0f                                                                  |     |
| Added                                                                                                     |               | General: Check for old format<br>modified /2319 . . . . .                                    | 37  |
| \textasteriskcentered . . . . .                                                                           | 479, 490      | 1996-11-23 ltoutenc.dtx v1.8e                                                                |     |
| Corrected syntax descriptions . . . . .                                                                   | 467           | General: Corrected description . . . . .                                                     | 467 |
| Removed \aa and \AA . . . . .                                                                             | 479, 483, 486 | Extended description . . . . .                                                               | 468 |
| 1996-10-28 lplain.dtx v1.1u                                                                               |               | 1996-11-28 ltvers.dtx v1.0g                                                                  |     |
| General: (CAR) More doc changes . . . . .                                                                 | 14            | General: Check for old format<br>modified /2319 . . . . .                                    | 37  |
| \dotfill: Removed math mode . . . . .                                                                     | 32            | 1996-12-06 ltdirchk.dtx v1.0u                                                                |     |
| 1996-10-29 lplain.dtx v1.1v                                                                               |               | \IfFileExists: *** removed from<br>various messages for GNU Make.<br>internal/2338 . . . . . | 10  |
| \dotfill: Got arithmetic correct<br>(CAR) . . . . .                                                       | 32            | 1996-12-06 ltfloor.dtx v1.1r                                                                 |     |
| 1996-10-29 lspace.dtx v1.2u                                                                               |               | \caption: Call \setminpage if<br>needed. latex/2318 . . . . .                                | 914 |
| \gnewline: Added macro . . . . .                                                                          | 427           | 1996-12-06 ltfssini.dtx v3.0h                                                                |     |
| \@no@lnbk: Macro replaces \lnbk and<br>\@nolnbk . . . . .                                                 | 425           | General: (DPC) Remove *** from<br>messages internal/2338 . . . . .                           | 660 |
| \@: Corrected and rationalised code . . . . .                                                             | 425           | 1996-12-17 ltdefns.dtx v1.0w                                                                 |     |
| \nolinebreak: Reimplemented both<br>using \no@lnbk . . . . .                                              | 424           | \g@addto@macro: Use \begingroup to<br>save making a mathord . . . . .                        | 112 |
| 1996-10-31 lfinal.dtx v1.0z                                                                               |               | 1996-12-20 ltsect.dtx v1.0z                                                                  |     |
| General: Added extra \lcode, hoping<br>it does no harm in T1<br>(pr/1969) . . . . .                       | 1180, 1187    | \@dottedtocline: Added \nobreak for<br>latex/2343 . . . . .                                  | 909 |
| 1996-10-31 ltlists.dtx v1.0p                                                                              |               | 1997-01-08 fontdef.dtx v2.2q                                                                 |     |
| \@trivlist: Added check for missing<br>item in outer list . . . . .                                       | 809           | General: Use \DeclareMathDelimiter<br>to set delimiter codes . . . . .                       | 672 |
| 1996-10-31 ltsect.dtx v1.0y                                                                               |               | \mathparagraph: Define using<br>\DeclareMathSymbol . . . . .                                 | 680 |
| General: Corrected and tidied<br>documentation; removed long<br>lines . . . . .                           | 898           | 1997-01-08 ltfiles.dtx v1.1j                                                                 |     |
| 1996-11-03 lplain.dtx v1.1w                                                                               |               | \cinclude: reset \deadcycles<br>latex/2365 . . . . .                                         | 453 |
| \dotfill: Saved tokens by using<br>\hb@xt@ . . . . .                                                      | 32            | 1997-01-08 ltmath.dtx v1.1d                                                                  |     |
| 1996-11-04 lterror.dtx v1.2m                                                                              |               | \root: (DPC) Remove spurious space<br>tokens from plain TeX definition<br>/2359 . . . . .    | 787 |
| \@nodocument: Always define<br>\@nodocument in kernel, so that it<br>can be cleared by \document. . . . . | 395           | 1997-02-05 ltdefns.dtx v1.0x                                                                 |     |
| 1996-11-04 ltlists.dtx v1.0q                                                                              |               | \g@addto@macro: missing percent<br>/2402 . . . . .                                           | 112 |
| \@trivlist: Moved check for missing<br>item: only checked when not<br>inlabel flag is false . . . . .     | 809           | 1997-02-21 ltlists.dtx v1.0r                                                                 |     |
| 1996-11-05 ltfiles.dtx v1.1i                                                                              |               | \@item: \ifvoid check added for<br>\noindent. latex/2414 . . . . .                           | 814 |
| \nofiles: Standard \if@nobreak test<br>added . . . . .                                                    | 449           | 1997-03-21 ltcounts.dtx v1.1e                                                                |     |
| 1996-11-09 ltmath.dtx v1.1c                                                                               |               | \fnsymbol: Use \mathsection and<br>\mathparagraph. latex/2445 . . . . .                      | 518 |
| \@ensuredmath: Made long, as it was<br>before. /2104 . . . . .                                            | 797           |                                                                                              |     |
| 1996-11-18 ltffssbas.dtx v3.0s                                                                            |               |                                                                                              |     |
| \define@newfont: (DPC) lowercase fd<br>file names. internal/1044 . . . . .                                | 538           |                                                                                              |     |

|                                                                                                                                           |      |                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------|
| 1997-04-14 ltfiles.dtx v1.1k                                                                                                              |      | 1997-08-29 ltoutenc.dtx v1.9f                                                                                           |
| \document: Set the document space factor defaults. latex/2404 . . . . .                                                                   | 445  | General: Added OT4 encoding, provided by Marcin Woliński. . . . .                                                       |
| \normalsfcodes: Macro added (from patch file) latex/2404 . . . . .                                                                        | 449  |                                                                                                                         |
| 1997-04-14 ltoutput.dtx v1.2b                                                                                                             |      | 1997-09-09 ltdefns.dtx v1.2z                                                                                            |
| \@writesetup: Call \normalsfcodes (from patch file) latex/2404 . . . . .                                                                  | 1116 | \provide@command: Use \begingroup to avoid generating math ords if used in math mode. pr/2573 . . . . .                 |
| Move \label and \index (from patch file) . . . . .                                                                                        | 1116 | 88                                                                                                                      |
| 1997-04-24 ltbibl.dtx v1.1m                                                                                                               |      | 1997-09-15 ltpictur.dtx v1.1g                                                                                           |
| \@citex: \@empty to avoid primitive error on empty cite keys. latex/2432 . . . . .                                                        | 939  | \@getcirc: Warn if lines become invisible pr/2524 . . . . .                                                             |
| 1997-04-30 ltoutenc.dtx v1.9a                                                                                                             |      | 883                                                                                                                     |
| General: Changed \textsc to \scshape . . . . .                                                                                            | 480  | \@picture@warn: Macro added pr/2524 . . . . .                                                                           |
| Introduced \textcopyright and modified \copyright . . . . .                                                                               | 480  | 883                                                                                                                     |
| Introduced \textcopyright and modify \copyright . . . . .                                                                                 | 481  | \@sline: Warn if lines become invisible pr/2524 . . . . .                                                               |
| Modified \textunderscore, removing \mathunderscore . . . . .                                                                              | 480  | 872                                                                                                                     |
| Modified \underscore, removing \mathunderscore . . . . .                                                                                  | 481  | 1997-10-06 lccounts.dtx v1.1f                                                                                           |
| 1997-04-30 ltoutenc.dtx v1.9b                                                                                                             |      | \@Roman: Change \@Roman to be fully expandable, so that the result is written properly to files. . . . .                |
| General: Added \leavevmode to \textunderscore . . . . .                                                                                   | 480  | 518                                                                                                                     |
| 1997-05-04 ltoutenc.dtx v1.9c                                                                                                             |      | \@slowromancap: Macro added. . . . .                                                                                    |
| General: Added ‘hex index tabs’ . . . . .                                                                                                 | 487  | 518                                                                                                                     |
| Added TS1 encoding v2.2.beta . . . . .                                                                                                    | 493  | 1997-10-08 ltlogos.dtx v1.1h                                                                                            |
| 1997-05-07 ltoutenc.dtx v1.9d                                                                                                             |      | \LaTeX: Simplify macro (force loading of suitable math fonts once). . . . .                                             |
| General: Added \leavevmode to \textcompwordmark . . . . .                                                                                 | 480  | 441                                                                                                                     |
| 1997-05-07 ltspace.dtx v1.2v                                                                                                              |      | 1997-10-10 ltclass.dtx v1.0y                                                                                            |
| \newline: Made completely robust. . . . .                                                                                                 | 426  | \endfilecontents: \@currenvir in banner . . . . .                                                                       |
| 1997-05-29 lfsstrc.dtx v3.0j                                                                                                              |      | 1012                                                                                                                    |
| General: Replaced \\ by \MessageBreak, as suggested by Donald Arseneau. . . . .                                                           | 574  | \reserved@c not \verb@out to save a csname . . . . .                                                                    |
| 1997-05-29 ltlogos.dtx v1.1f                                                                                                              |      | 1011                                                                                                                    |
| \LaTeXe: Added \math so that the L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> logo works with non-zero values of \mathsurround. . . . . | 441  | Check for text before or after \end environment. latex/2636 . . . . .                                                   |
| 1997-06-16 ltdirchk.dtx v1.0v                                                                                                             |      | 1012                                                                                                                    |
| General: documentation improvements mainly from internal/2520 . . . . .                                                                   | 1    | Use \@gobbletwo . . . . .                                                                                               |
| 1997-06-16 ltfloat.dtx v1.1s                                                                                                              |      | 1011                                                                                                                    |
| General: documentation fixes . . . . .                                                                                                    | 911  | 1997-10-17 lfntcmd.dtx v3.3w                                                                                            |
| 1997-06-16 lfntcmd.dtx v3.3v                                                                                                              |      | \check@nocorr@: Check for vertical mode moved here, from \DeclareTextFontCommand (see PR/2646). . . . .                 |
| General: Fix typo in documentation. . . . .                                                                                               | 686  | 690                                                                                                                     |
| 1997-08-05 ltoutenc.dtx v1.9e                                                                                                             |      | \DeclareTextFontCommand:                                                                                                |
| General: Corrected order of arguments in \UseTextSymbol example. . . . .                                                                  | 467  | Reinstalled \check@icr as check is now done in \check@nocorr@ (see PR/2646). . . . .                                    |
|                                                                                                                                           |      | 688                                                                                                                     |
|                                                                                                                                           |      | 1997-10-20 ltfinal.dtx v1.1a                                                                                            |
|                                                                                                                                           |      | \cuclclist: Removed \aa and \AA from \cuclclist as these are macros. . . . .                                            |
|                                                                                                                                           |      | 1188                                                                                                                    |
|                                                                                                                                           |      | 1997-10-21 ltdefns.dtx v1.2z1                                                                                           |
|                                                                                                                                           |      | \renew@command: Use \begingroup/\endgroup rather than braces for grouping, to avoid generating empty math atom. . . . . |
|                                                                                                                                           |      | 86                                                                                                                      |
|                                                                                                                                           |      | 1997-10-21 lfssbas.dtx v3.0t                                                                                            |
|                                                                                                                                           |      | \define@newfont: Move \makeatletter to \nfss@catcodes. . . . .                                                          |
|                                                                                                                                           |      | 537                                                                                                                     |

|                                                                                                                                                                                              |      |                                                                                                             |      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------|------|
| \nfss@catcodes: Moved<br>\makeatletter from<br>\try@load@font@shape. . . . .                                                                                                                 | 538  | Removed default settings, see next<br>section. . . . .                                                      | 493  |
| 1997-11-09 ltoutput.dtx v1.2c<br>\@specialoutput: Remove incorrect<br>code: only one \emptycol is<br>needed here . . . . .                                                                   | 1105 | 1997-12-19 ltoutenc.dtx v1.9i<br>General: Documentation corrections. 466                                    |      |
| \@topnewpage: Documentation of vsize<br>check enhanced . . . . .                                                                                                                             | 1102 | 1997-12-20 fontdef.dtx v2.2s<br>General: Added documentation . . . . .                                      | 664  |
| 1997-11-13 ltfsdcl.dtx v3.0f<br>\DeclareSymbolFont: (DPC) Really<br>update \group@list don't leave<br>new version in \toks@. latex/2661 615                                                  |      | 1997-12-31 ltoutenc.dtx v1.9k<br>General: Further correction . . . . .                                      | 467  |
| \stepcounter: (DPC) Remove as<br>never used. (Re)defined in<br>ltcounts . . . . .                                                                                                            | 604  | 1998-01-12 ltoutenc.dtx v1.9k<br>General: Added \ProvidesPackage for<br>textcomp.sty . . . . .              | 466  |
| 1997-11-19 ltfloat.dtx v1.1t<br>\@footnotetext: Missing percent,<br>again . . . . .                                                                                                          | 930  | Adding missing braces and<br>\ushape. . . . .                                                               | 495  |
| 1997-11-19 ltoutput.dtx v1.2d<br>\@vtryfc: Reindent code, to be<br>understandable(DPC). . . . .                                                                                              | 1122 | 1998-01-16 ltoutenc.dtx v1.9m<br>General: fixed decimal codes.<br>latex/2734 . . . . .                      | 491  |
| 1997-11-20 ltfsdcl.dtx v3.0g<br>\document@select@group: (DPC)<br>inline use of \stepcounter (faster,<br>and saves a csname per math<br>version as no reset list) . . . . .                   | 607  | 1998-03-04 ltdefns.dtx v1.2z2<br>\@xargdef: Unnecessary<br>\expandafter removed: pr/2758 . 84               |      |
| \select@group: (DPC) inline use of<br>\stepcounter (faster, and saves a<br>csname per math version as no<br>reset list) . . . . .                                                            | 605  | 1998-03-05 ltoutenc.dtx v1.9n<br>General: Added masc/fem ords as in<br>pr/2579 . . . . .                    | 480  |
| 1997-11-23 ltoutenc.dtx v1.9g<br>General: Use \textperthousand,<br>\textpertenthousand and<br>\textfractionsolidus not<br>\textpermill, \textperenmill<br>and \textfraction. /2673 . . . . . | 493  | 1998-03-20 ltdefns.dtx v1.2z3<br>\@thirdofthree: Macro added . . . . .                                      | 89   |
| 1997-12-17 ltoutenc.dtx v1.9h<br>General: Added \textperthousand<br>and \textpertenthousand . 484, 485                                                                                       |      | 1998-03-20 ltoutenc.dtx v1.9o<br>General: Documentation added about<br>order of decls . . . . .             | 469  |
| Added code for textcomp.sty. . . . .                                                                                                                                                         | 509  | Documentation added for pr/2783 468                                                                         |      |
| Added section. . . . .                                                                                                                                                                       | 509  | \UndeclareTextCommand: Macro added<br>for pr/2783 . . . . .                                                 | 477  |
| Added textcomp.sty. . . . .                                                                                                                                                                  | 466  | 1998-03-20 lttextcomp.dtx v1.9o<br>General: Added various                                                   |      |
| As in OT1, Added \leavevmode at<br>start of \c, otherwise the output<br>routine might be invoked within<br>the macro. . . . .                                                                | 485  | \UndeclareTextCommand<br>declarations for pr/2783 . . . . .                                                 | 725  |
| Changed to decimal codes in<br>\ooalign. . . . .                                                                                                                                             | 495  | Load decls after defaults for speed. 725                                                                    |      |
| Changed to decimal codes. . . . .                                                                                                                                                            | 491  | 1998-03-21 ltclass.dtx v1.0z<br>General: Added to documentation of                                          |      |
| Documentation changes and<br>additions. . . . .                                                                                                                                              | 466  | filecontents . . . . .                                                                                      | 974  |
| Example corrected, braces<br>removed. . . . .                                                                                                                                                | 466  | 1998-03-21 ltclass.dtx v1.1a<br>\@providesfile: Allow &<br>Internal/2702 . . . . .                          | 989  |
|                                                                                                                                                                                              |      | General: Correct to new onlypreamble<br>command list . . . . .                                              | 1026 |
|                                                                                                                                                                                              |      | 1998-03-25 ltfsbas.dtx v3.0u<br>\showhyphens: Suppress unnecessary<br>error when used in preamble . . . . . | 547  |
|                                                                                                                                                                                              |      | 1998-04-11 fontdef.dtx v2.2t<br>General: Added \mathring accent<br>(pr2785) . . . . .                       | 678  |
|                                                                                                                                                                                              |      | 1998-04-15 fontdef.dtx v2.2u<br>General: Use new syntax for<br>\DeclareMathDelimiter . . . . .              | 672  |

|                                                                                                            |      |                                                                                                                  |     |
|------------------------------------------------------------------------------------------------------------|------|------------------------------------------------------------------------------------------------------------------|-----|
| 1998-04-15 ltfssdcl.dtx v3.0h                                                                              |      | 1998-08-17 ltdirchk.dtx v1.0w                                                                                    |     |
| \@xxDeclareMathDelimiter: Macro<br>added (pr/2662) . . . . .                                               | 626  | General: (RmS) Documentation<br>improvements. . . . .                                                            | 1   |
| 1998-04-17 fontdef.dtx v2.2v                                                                               |      | 1998-08-17 lfntcmd.dtx v3.3x                                                                                     |     |
| General: Reinsert symbol defs for <<br>and > chars. . . . .                                                | 672  | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 686 |
| 1998-04-18 fontdef.dtx v2.2w                                                                               |      | 1998-08-17 lfssbas.dtx v3.0v                                                                                     |     |
| General: Reinsert symbol def for /<br>char. . . . .                                                        | 672  | General: (RmS) Documentation fixes. . . . .                                                                      | 523 |
| 1998-05-07 ltclass.dtx v1.1b                                                                               |      | 1998-08-17 lfssdcl.dtx v3.0i                                                                                     |     |
| \@onefilewithoptions@clashchk:<br>Modify help message for<br>latex/2805 . . . . .                          | 1003 | General: (RmS) Corrected minor<br>glitches in changes entries. . . . .                                           | 602 |
| 1998-05-18 lttab.dtx v1.1j                                                                                 |      | 1998-08-17 lfssini.dtx v3.0i                                                                                     |     |
| \@endpbox: Use \setlength to set<br>\hsize, so that the changes in the<br>calc package apply here. . . . . | 860  | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 634 |
| \@tabular*: Use \setlength, so that<br>calc extensions apply. . . . .                                      | 849  | 1998-08-17 ltlogos.dtx v1.1i                                                                                     |     |
| 1998-05-20 ltfinal.dtx v1.1b                                                                               |      | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 441 |
| General: Set up lccodes before loading<br>hyphenation files: pr/2639 . . . . .                             | 1179 | 1998-08-17 ltmath.dtx v1.1c                                                                                      |     |
| Set up uc/lccodes after loading<br>hyphenation files: pr/2639 . . . . .                                    | 1187 | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 785 |
| 1998-05-28 lterror.dtx v1.2n                                                                               |      | 1998-08-17 ltmiscen.dtx v1.1g                                                                                    |     |
| \@notdefinable: Added message re<br>‘end..’ pr/1555 . . . . .                                              | 395  | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 760 |
| 1998-06-04 ltboxes.dtx v1.1c                                                                               |      | 1998-08-17 ltspace.dtx v1.2w                                                                                     |     |
| \@rule: Support calc-expressions . . .                                                                     | 834  | General: Documentation fixes. . . . .                                                                            | 421 |
| 1998-06-12 ltoutenc.dtx v1.9p                                                                              |      | 1998-08-17 preload.dtx v2.1g                                                                                     |     |
| General: Corrected 130 and 131, see<br>pr/2834 . . . . .                                                   | 496  | General: (RmS) Minor documentation<br>fixes. . . . .                                                             | 683 |
| Renamed \textmacron pr/2840 . . . . .                                                                      | 497  | 1998-09-19 ltoutenc.dtx v1.9r                                                                                    |     |
| 1998-06-12 ltoutenc.dtx v1.9q                                                                              |      | \@a: Added \string (pr/2878) . . . . .                                                                           | 478 |
| \add@accent: Explicitly set<br>\spacefactor after \accent<br>(pr/2877) . . . . .                           | 473  | 1998-11-13 lttab.dtx v1.1m                                                                                       |     |
| 1998-06-12 lttextcomp.dtx v1.9p                                                                            |      | \@array: Check for hmode to see if<br>something went wrong during<br>parsing (pr/2884) . . . . .                 | 850 |
| General: Renamed \textmacron<br>pr/2840 . . . . .                                                          | 721  | 1999-01-05 fontdef.dtx v2.2x                                                                                     |     |
| 1998-06-18 lttab.dtx v1.1k                                                                                 |      | General: Need special protection for<br>character > in \changes entry. . . . .                                   | 662 |
| General: Small addition to<br>documentation . . . . .                                                      | 837  | 1999-01-06 lfssbas.dtx v3.0w                                                                                     |     |
| 1998-07-06 lttab.dtx v1.1l                                                                                 |      | \@DeclareFontEncoding: Added<br>\LastDeclaredEncoding to<br>support cyrillic integration<br>(pr/2988) . . . . .  | 527 |
| General: Small correction to<br>documentation . . . . .                                                    | 837  | \@LastDeclaredEncoding: Added<br>\LastDeclaredEncoding to<br>support cyrillic integration<br>(pr/2988) . . . . . | 527 |
| 1998-08-17 ltboxes.dtx v1.1e                                                                               |      | 1999-01-06 ltoutenc.dtx v1.9r                                                                                    |     |
| General: (RmS) Minor Documentation<br>fixes. . . . .                                                       | 818  | \@strip@args: New impl for<br>latex/2930 . . . . .                                                               | 475 |
| 1998-08-17 ltclass.dtx v1.1c                                                                               |      | General: Minor documentation fix. . . . .                                                                        | 495 |
| General: (RmS) Minor documentation<br>fixes. . . . .                                                       | 974  | 1999-01-06 ltoutput.dtx v1.2e                                                                                    |     |
|                                                                                                            |      | \@makecol: Added negative vskip, as<br>when processing outputbox below:                                          |     |

|                                                                                                                                       |          |                                                                                                                                                                                                              |          |
|---------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| suggested by Fred Bartlett<br>pr/2892 . . . . .                                                                                       | 1111     | New macro added . . . . .                                                                                                                                                                                    | 85       |
| 1999-01-07 ltdefns.dtx v1.3a<br>\@ifnextchar: made long . . . . .                                                                     | 107      | 1999-06-10 ltoutenc.dtx v1.9u<br>General: Ensure that we also forget<br>old options (pr/2888) . . . . .                                                                                                      | 511      |
| \@newenvb: made long and brace<br>optional arg. latex/2896 . . . . .                                                                  | 87       | 1999-06-12 ltoutenc.dtx v1.9v<br>General: Extend \@uclclist only<br>once . . . . .                                                                                                                           | 510      |
| \@testopt: made long and brace<br>optional arg. latex/2896 . . . . .                                                                  | 84       | 1999-10-09 ltmath.dtx v1.1e<br>\active@math@prime: Macro added,<br>see PR 3104. . . . .                                                                                                                      | 792      |
| 1999-01-07 ltdefns.dtx v1.3b<br>\@ifnextchar: extra \long.<br>latex/2902 . . . . .                                                    | 107      | \prime@s: Introduce<br>\active@math@prime. . . . .                                                                                                                                                           | 792      |
| 1999-01-07 ltoutenc.dtx v1.9r<br>General: Hackery to allow using<br>fontenc several times . . . . .                                   | 511      | 1999-10-09 ltoutput.dtx 1.2f<br>\@activechar@info: Reset definition<br>of active prime character (used in<br>math mode) . . . . .                                                                            | 1114     |
| Hackery to temp support cyrillic<br>uc/lc . . . . .                                                                                   | 509      | 1999-10-28 ltoutenc.dtx v1.9w<br>\add@accent: Give<br>\accent@spacefactor a default<br>definition (pr/3084) . . . . .                                                                                        | 473      |
| 1999-01-13 ltoutenc.dtx v1.9s<br>\@strip@args: Simplified solution for<br>latex/2930 . . . . .                                        | 475      | 1999-12-08 ltoutenc.dtx v1.9x<br>General: Changed \CYRRHOOK and<br>\cyrrhook to \CYRRHK and \cyrrhk<br>as name changed in the cyrillic<br>bundle for naming consistency<br>with other "hook" glyphs. . . . . | 509      |
| 1999-01-18 ltdefns.dtx v1.3c<br>\@yargd@f: New implementation DPC<br>/2942 . . . . .                                                  | 85       | 2000-01-07 ltmiscen.dtx v1.1h<br>\@verbatim: Disable hyphenation even<br>if the font allows it. . . . .                                                                                                      | 778      |
| 1999-02-09 ltdefns.dtx v1.3d<br>\@yargd@f: catch bad argument forms<br>by re-inserting #3 . . . . .                                   | 85       | 2000-01-15 ltpictur.dtx v1.1i<br>\@upvector: Removed space at<br>end-of-line, CAR . . . . .                                                                                                                  | 875      |
| 1999-02-12 lttextcomp.dtx v3.0j<br>\legacyoldstylenums: Use<br>\rmdefault instead of cmm<br>(pr/2954) . . . . .                       | 695      | 2000-01-30 ltfntcmd.dtx v3.3y<br>\DeclareTextFontCommand: Use<br>\hmode@bgroup now (pr/3160) . . .                                                                                                           | 688      |
| 1999-02-24 ltoutenc.dtx v1.9t<br>General: Corrected hackery cyrillic<br>uc/lc list . . . . .                                          | 509      | 2000-01-30 ltoutenc.dtx v1.9y<br>General: Use \hmode@bgroup where<br>applicable<br>(pr/3160) . . . 482–485, 490–493, 495                                                                                     |          |
| 1999-03-01 ltdefns.dtx v1.3e<br>\@ifnextchar: remove extra \long.<br>internal/2967 . . . . .                                          | 107      | \add@accent: Use \hmode@bgroup<br>where applicable (pr/3160) . . . . .                                                                                                                                       | 472      |
| 1999-04-15 ltpictur.dtx v1.1h<br>\@getlarrow: Replaced octal number,<br>CAR . . . . .                                                 | 874      | \hmode@bgroup: Macro added . . . . .                                                                                                                                                                         | 473      |
| \@upvector: Replaced octal number,<br>CAR . . . . .                                                                                   | 875      | 2000-01-30 ltoutenc.dtx v1.9z<br>\@use@text@encoding: Macro<br>reimplemented (pr/3160) . . . . .                                                                                                             | 475, 476 |
| General: Replaced octal number,<br>CAR . . . . .                                                                                      | 874, 875 | \add@accent: Macro reimplemented<br>(pr/3160) . . . . .                                                                                                                                                      | 472      |
| Replaced octal numbers, CAR . . .                                                                                                     | 862      | \hmode@start@before@group: Macro<br>added (pr/3160) . . . . .                                                                                                                                                | 476      |
| 1999-04-19 ltfloat.dtx v1.1u<br>\caption: Made caption an error<br>outside a float: latex/2815 . . . . .                              | 914      | 2000-05-19 ltmiscen.dtx v1.1i<br>\enddocument: Reset \AtEndDocument<br>for latex/3060 . . . . .                                                                                                              | 761      |
| 1999-04-27 ltboxes.dtx v1.1f<br>\@parboxto: (CAR) Changed \empty<br>to \relax as flag for natural<br>width: pr/2975 . . . . .         | 828      |                                                                                                                                                                                                              |          |
| 1999-04-29 ltdefns.dtx v1.3f<br>\@yargd@f: Full expansion and<br>conversion needed for digit in new<br>version, see pr/3013 . . . . . | 85       |                                                                                                                                                                                                              |          |

|                               |                                                                                                      |      |                               |                                                                                       |     |
|-------------------------------|------------------------------------------------------------------------------------------------------|------|-------------------------------|---------------------------------------------------------------------------------------|-----|
| 2000-05-26 ltpage.dtx v1.0j   | \@markright: Reimplementation to fix expansion error (pr/3203). . . . .                              | 972  | 2001-02-16 ltxref.dtx v1.1k   | \@newl@bel: Added an extra grouplevel (PR3250), jlb . . . . .                         | 741 |
|                               | \leftmark: Use \empty instead of brace group (pr/3203). . . . .                                      | 972  | 2001-05-25 ltclass.dtx v1.1d  | \@providesfile: Explicitly set catcode of \endlinechar to 10 (pr/3334) . . . . .      | 989 |
|                               | \markright: Reimplementation to fix expansion error (pr/3203). . . . .                               | 970  | 2001-05-25 ltdirchk.dtx v1.0x | General: Explicitly set catcode of \endlinechar to 10 (pr/3334) . . . . .             | 4   |
|                               | \rightmark: Use \empty instead of brace group (pr/3203). . . . .                                     | 972  | 2001-05-28 ltoutenc.dtx v1.93 | General: Added composites for compatibility with T1, pr/3295 . . . . .                | 484 |
| 2000-06-02 ltpage.dtx v1.0k   | \@markright: Small adjustment to give slightly less expansion, CAR                                   | 972  |                               | Changed the effect of \.\i, pr/3295 . . . . .                                         | 487 |
|                               | \markright: Small adjustment to give slightly less expansion, CAR . . . . .                          | 970  | 2001-06-02 fontdef.dtx v2.2y  | General: Provide default cfg files (pr/3264) . . . . .                                | 681 |
|                               | Tidied 1.0j reimplementation, CAR                                                                    | 970  | 2001-06-04 fontdef.dtx v2.2z  | General: Guard against math active equal and pipe sign in \models (pr/3333) . . . . . | 677 |
| 2000-07-11 ltmiscen.dtx v1.1j | \@enddocument@kernel@warnings: Fix typo in warning . . . . .                                         | 763  |                               | Guard against math active equal sign in \Relbar (pr/3333) . . . . .                   | 677 |
| 2000-07-12 ltoutput.dtx 1.2g  | General: Ensure that rule is in \normalcolor . . . . .                                               | 1155 | 2001-06-04 ltclass.dtx v1.1e  | \@providesfile: But only if it is a char (pr/3334) . . . . .                          | 989 |
|                               | \@makecol: Removed negative vskip, as it gives unacceptable results when the depth is large: pr/3189 | 1111 | 2001-06-04 ltdirchk.dtx v1.0y | General: But only if it is a char (pr/3334) . . . . .                                 | 4   |
| 2000-07-19 ltoutput.dtx v1.2h | \@writesetup: Reset and restore \@if@newlist for internal/3231                                       | 1115 | 2001-06-04 ltpictur.dtx v1.1j | \@sline: Don't warn for exactly zero pr/3318 . . . . .                                | 872 |
| 2000-08-23 ltfinal.dtx v1.1c  | General: Fix typo in warning . . . . .                                                               | 1181 | 2001-06-04 ltvers.dtx v1.0i   | General: Check for old format disabled                                                | 37  |
| 2000-08-30 ltoutenc.dtx v1.91 | \use@text@encoding: Rearranged but no change to final code, CAR (pr/3160) . . . . .                  | 475  | 2001-06-05 ltoutenc.dtx v1.94 | General: Text composite Commands need kludges for ',' – see tlb1903.lvt . . . . .     | 484 |
|                               | \add@accent: Rearranged but no change to final code, CAR (pr/3160) . . . . .                         | 472  | 2001-08-26 ltclass.dtx v1.1f  | \@providesfile: Readded setting of space char (pr/3353) . . . . .                     | 989 |
| 2000-09-01 ltfinal.dtx v1.1d  | \errhelp: Set error help empty at very end (pr/449 done correctly). . . . .                          | 1191 | 2002-02-24 lplain.dtx v1.1x   | \loggingall: Macro added . . . . .                                                    | 33  |
|                               | \end@dblfloat: FMi: use output routine to defer float . . . . .                                      | 919  |                               | \loggingoutput: Macro added . . . . .                                                 | 33  |
| 2000-09-24 ltfloat.dtx v1.2b  | \docclearpage: FMi: ensure \docclearpage is called again until all floats are output. . . . .        | 1109 |                               | \showoutput: Use newly added \loggingoutput . . . . .                                 | 33  |
|                               | \@addtocurcol: FMi: test for wide float was in wrong place . . . . .                                 | 1128 |                               | \tracingall: Use newly added \loggingoutput . . . . .                                 | 33  |
| 2001-01-07 ltoutput.dtx v1.2j | \@writesetup: And do it in the right macro (pr/3286) . . . . .                                       | 1115 | 2002-06-16 ltoutenc.dtx v1.95 | General: Added \textbardbl (pr/3400) . . . . .                                        | 490 |
|                               |                                                                                                      |      |                               | Added default for \textbardbl (pr/3400) . . . . .                                     | 479 |

|                                                                                                                                                                                                            |          |                                                                        |      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|------------------------------------------------------------------------|------|
| 2002-06-17 ltoutenc.dtx v1.95                                                                                                                                                                              |          | 2004-01-03 ltoutenc.dtx v1.99b                                         |      |
| General: Corrected \c for T1<br>(pr/3442) .....                                                                                                                                                            | 485      | General: Added \textogonekcentered<br>(pr/3532) .....                  | 485  |
| Definition of \texttexclamdown<br>changed (pr/3368) .....                                                                                                                                                  | 483      | Added composites for \k (pr/3532) 490                                  |      |
| Definition of \textquestiondown<br>changed (pr/3368) .....                                                                                                                                                 | 483      | Use \oalign for \k (pr/3532) ... 485                                   |      |
| 2002-06-18 ltoutenc.dtx v1.95                                                                                                                                                                              |          | 2004-01-04 ltbibl.dtx v1.1p                                            |      |
| General: Changed def for<br>\textregistered to avoid small<br>caps (pr/3420) .....                                                                                                                         | 480      | \nocite: Changed error message ... 940                                 |      |
| 2002-10-01 ltfloat.dtx v1.1v                                                                                                                                                                               |          | 2004-01-04 ltoutenc.dtx v1.99c                                         |      |
| \thempfootnote: Use braces around<br>\itshape to keep font change local<br>(pr/3460). .....                                                                                                                | 928      | General: More adjustments for ogonek<br>(pr/3532) .....                | 485  |
| 2002-10-02 ltfssbas.dtx v3.0x                                                                                                                                                                              |          | 2004-01-23 ltdefns.dtx v1.1g                                           |      |
| \DeclareFontSubstitution: Adding<br>\LastDeclaredEncoding<br>introduced a bug as on some<br>occasions that macro name was<br>stored in the internal lists instead<br>of the actual encoding. (pr/3459) 527 |          | \newenva: Use kernel version of<br>\ifnextchar (pr/3501) .....         | 87   |
| 2002-10-28 ltlists.dtx v1.0s                                                                                                                                                                               |          | \testopt: Use kernel version of<br>\ifnextchar (pr/3501) .....         | 84   |
| \endtrivlist: Check for math mode<br>(pr/3437) .....                                                                                                                                                       | 810      | \xargdef: Use kernel version of<br>\ifnextchar (pr/3501) .....         | 84   |
| 2002-10-28 ltoutenc.dtx v1.96                                                                                                                                                                              |          | \xdblarg: Use kernel version of<br>\ifnextchar (pr/3501) .....         | 108  |
| General: coding change, to follow bug<br>fix by DEK in plain.tex<br>(pr/3469) .....                                                                                                                        | 483, 492 | 2004-01-23 ltdefns.dtx v1.3g                                           |      |
| 2002-12-13 ltbibl.dtx v1.1n                                                                                                                                                                                |          | \kernel@ifnextchar: Added macro<br>(pr/3501) .....                     | 107  |
| \citex: Added \leavevmode in case<br>citation is at start of paragraph<br>(pr/3486) .....                                                                                                                  | 939      | 2004-01-28 ltclass.dtx v1.1g                                           |      |
| 2003-01-01 ltfntcmd.dtx v3.3z                                                                                                                                                                              |          | \providesfile: Use kernel version of<br>\ifnextchar (pr/3501) .....    | 989  |
| General: Code checked and<br>documentation extended by Chris 688                                                                                                                                           |          | 2004-01-28 ltvers.dtx v1.0k                                            |      |
| 2003-05-18 ltbibl.dtx v1.1o                                                                                                                                                                                |          | General: Check for old format made 5<br>years (pr/3601) .....          | 37   |
| \nocite: Check if we are after<br>\document .....                                                                                                                                                          | 940      | 2004-02-02 fontdef.dtx v2.3                                            |      |
| 2003-08-27 ltpictur.dtx v1.1k                                                                                                                                                                              |          | General: Many things from here on<br>made robust .....                 | 676  |
| \bezier: added missing displacement<br>pr/3566 .....                                                                                                                                                       | 891      | 2004-02-02 ltoutenc.dtx v1.99                                          |      |
| \sline: check for \linechar being<br>empty pr/3570 .....                                                                                                                                                   | 872      | General: Added \textbigcircle ... 490                                  |      |
| 2003-10-13 ltfinal.dtx v1.1e                                                                                                                                                                               |          | 2004-02-04 fontdef.dtx v2.3a                                           |      |
| General: Added extra \lccode for \-                                                                                                                                                                        |          | General: Added bigtriangle synonyms<br>for stmaryrd .....              | 674  |
| and \textcompwordmark .....                                                                                                                                                                                | 1180     | 2004-02-04 ltspace.dtx v1.3                                            |      |
| 2003-12-16 ltoutput.dtx v1.2k                                                                                                                                                                              |          | \nobreakdashes: Macro added .... 437                                   |      |
| \makecol: Ensure that \elt has a<br>defined state (pr/3586) .....                                                                                                                                          | 1111     | 2004-02-06 ltoutenc.dtx v1.99d                                         |      |
| 2003-12-30 ltpictur.dtx v1.1j                                                                                                                                                                              |          | \cinmathwarn: New command added<br>to fix severe bug: pr/3563 .... 470 |      |
| \getcirc: issue warning if circle size<br>can't be met pr/3473 .....                                                                                                                                       | 883      | 2004-02-07 ltoutput.dtx v1.2l                                          |      |
|                                                                                                                                                                                                            |          | \docclearpage: Empty kludgeins box<br>if necessary, pr/3528 .....      | 1109 |
|                                                                                                                                                                                                            |          | 2004-02-13 ltoutenc.dtx v1.99e                                         |      |
|                                                                                                                                                                                                            |          | General: Documentation fixes: typos 466                                |      |
|                                                                                                                                                                                                            |          | 2004-02-15 ltbibl.dtx v1.1q                                            |      |
|                                                                                                                                                                                                            |          | \cite@ofmt: Added hook with<br>default value \hbox .....               | 941  |
|                                                                                                                                                                                                            |          | \citex: Changed to use a hook with<br>default value \hbox .....        | 939  |
|                                                                                                                                                                                                            |          | 2004-02-15 ltspace.dtx v1.3a                                           |      |
|                                                                                                                                                                                                            |          | \nobreakdashes: Added spacefactor<br>setting .....                     | 437  |

|            |                             |                                                                                                                              |                                 |
|------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| 2004-10-20 | ltoutput.dtx                | v1.2m                                                                                                                        |                                 |
|            | \@makecol:                  | Removed dead code                                                                                                            | 1111                            |
| 2005-07-27 | ltfssdcl.dtx                | v3.0j                                                                                                                        |                                 |
|            | \DeclareMathAlphabet:       | (MH) Make document commands robust                                                                                           | 618                             |
|            | \DeclareSymbolFontAlphabet: | (MH) Make document commands robust                                                                                           | 631                             |
|            | \new@mathalphabet:          | (MH) Make document commands robust                                                                                           | 619                             |
|            | \non@alpherr:               | (MH) Change because command is now properly robust                                                                           | 606                             |
|            | \SetMathAlphabet:           | (MH) Make document commands robust                                                                                           | 620                             |
| 2005-09-27 | ltoutenc.dtx                | v1.99g                                                                                                                       |                                 |
|            | General:                    | Replace \sh@ft by \ltx@sh@ft                                                                                                 | 482, 485, 492                   |
| 2005-09-27 | ltplain.dtx                 | v1.1y                                                                                                                        |                                 |
|            | \ltx@sh@ft:                 | New macro                                                                                                                    | 32                              |
|            | \sh@ft:                     | Macro no longer used but left for compatibility                                                                              | 32                              |
| 2005-11-08 | ltoutenc.dtx                | v1.99h                                                                                                                       |                                 |
|            | General:                    | Added \ij and \IJ from babel.                                                                                                | pr/3771 . . . . . 479, 484, 486 |
| 2005-11-10 | ltmath.dtx                  | v1.1g                                                                                                                        |                                 |
|            | \[: (MH)                    | Fixed potential problem in \l (pr/3399).                                                                                     | 793                             |
|            | General:                    | (MH) Minor documentation fixes.                                                                                              | 785                             |
| 2006-05-18 | ltboxes.dtx                 | v1.1g                                                                                                                        |                                 |
|            | \@parboxto:                 | Ensure \@parboxto holds the value of \tempdimb not the register itself                                                       | pr/3867 . . . . . 828           |
| 2006-09-13 | ltoutput.dtx                | v1.1m                                                                                                                        |                                 |
|            | General:                    | Ensure that rule is in \normalcolor                                                                                          | 1156                            |
| 2007-08-05 | ltclass.dtx                 | v1.1h                                                                                                                        |                                 |
|            | \@fileswithoptions:         | Prevent loss of brackets PR/3965                                                                                             | 999                             |
| 2007-08-06 | ltcntrtl.dtx                | v1.0h                                                                                                                        |                                 |
|            | \@fornoop:                  | Really make defs long                                                                                                        | 387                             |
| 2007-08-31 | ltfssdcl.dtx                | v3.0l                                                                                                                        |                                 |
|            | \SetSymbolFont@:            | Font warning changed to info for encoding change                                                                             | pr/3975 . . . . . 617           |
| 2009-09-24 | ltvers.dtx                  | v1.0l                                                                                                                        |                                 |
|            | General:                    | Stop checking for old format                                                                                                 | 37                              |
| 2009-10-20 | ltfssdcl.dtx                | v3.0m                                                                                                                        |                                 |
|            | \in@:                       | More robust thanks to Heiko.                                                                                                 | 602                             |
| 2009-10-28 | lttextcomp.dtx              | v1.99k                                                                                                                       |                                 |
|            | General:                    | Added Latin Modern and TeX Gyre subsets                                                                                      | 726                             |
| 2009-11-04 | lttextcomp.dtx              | v1.99l                                                                                                                       |                                 |
|            | General:                    | Added more Latin Modern and TeX Gyre subsets                                                                                 | 726                             |
| 2009-12-14 | ltfntcmd.dtx                | v3.4a                                                                                                                        |                                 |
|            | \ifmaybe@ic:                | Macro added                                                                                                                  | 691                             |
|            | \maybe@ic@:                 | Use switch \ifmaybe@ic instead of \if@tempswa                                                                                | 691                             |
|            | \t@st@ic:                   | Use switch \ifmaybe@ic instead of \if@tempswa                                                                                | 692                             |
| 2010-08-17 | ltmiscen.dtx                | v1.1k                                                                                                                        |                                 |
|            | \enddocument:               | Use braces around \input arg (pr/4124)                                                                                       | 762                             |
| 2010-08-17 | ltmiscen.dtx                | v1.1l                                                                                                                        |                                 |
|            | \enddocument:               | Change of plan: use \@@input instead (pr/4124)                                                                               | 762                             |
| 2011-05-08 | ltfssdcl.dtx                | v3.0n                                                                                                                        |                                 |
|            | \in@:                       | Simplified thanks to Bruno.                                                                                                  | 602                             |
| 2011-08-19 | ltclass.dtx                 | v1.1i                                                                                                                        |                                 |
|            | \@ifclasswith:              | Re-jig definition after more stringent \in@ test                                                                             | 985                             |
| 2011-09-03 | ltfssdcl.dtx                | v3.0o                                                                                                                        |                                 |
|            | \new@mathversion:           | (Will) Remove \global before \newcount (unnecessary and caused etex bug).                                                    | 614                             |
| 2012-01-20 | ltplain.dtx                 | v2.0b                                                                                                                        |                                 |
|            | \loggingall:                | etex tracing if available                                                                                                    | 33                              |
| 2013-07-07 | ltclass.dtx                 | v1.1i                                                                                                                        |                                 |
|            | General:                    | Correctly describe how the date in \@ifpackagelater is used                                                                  | 977                             |
| 2014-04-18 | ltoutput.dtx                | v1.1o                                                                                                                        |                                 |
|            | General:                    | Handle infinite glue from \enlargethispage                                                                                   | pr/4023 . . . . . 1156          |
| 2014-04-24 | ltoutput.dtx                | v1.2n                                                                                                                        |                                 |
|            | \fl@tracemessage:           | Renamed internal trace commands; provide as package                                                                          | 1146                            |
| 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 | 919                             |
| 2014-06-10 | ltfloat.dtx                 | v1.2b                                                                                                                        |                                 |
|            | \end@dblfloat:              | missing \fi added                                                                                                            | 919                             |
| 2014-12-30 | ltfinal.dtx                 | v2.0a                                                                                                                        |                                 |
|            | \newmarks:                  | macro added                                                                                                                  | 1174                            |
|            | \newXeTeXintercharclass:    | macro added                                                                                                                  | 1175                            |
| 2014-12-30 | ltfloat.dtx                 | v1.2a                                                                                                                        |                                 |
|            | \@textsubscript:            | Command added (latexrelease)                                                                                                 | 930                             |
|            | \textsubscript:             | Command added (latexrelease)                                                                                                 | 929                             |
| 2014-12-30 | ltfssbas.dtx                | v3.0y                                                                                                                        |                                 |
|            | \mathgroup:                 | move allocation to ltplain.                                                                                                  | 523                             |

|                                                                                           |                                                                                              |
|-------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 2014-12-30 ltoutput.dtx v1.2m                                                             | 2015-01-10 lccounts.dtx v1.1h                                                                |
| General: Command updated<br>(latexrelease) . . . . .                                      | \@fnnsymbol: Unse \TextOrMath<br>(latexrelease) . . . . .                                    |
| 1155                                                                                      | 519                                                                                          |
| 2014-12-30 lplain.dtx v2.0a                                                               | \@stpelt: Reset all within counters in<br>one go (latexrelease) . . . . .                    |
| \@alloc: macro added . . . . .                                                            | 514                                                                                          |
| \@alloc@chardef: macro added . . . . .                                                    |                                                                                              |
| \@alloc@top: macro added . . . . .                                                        | 18                                                                                           |
| \@ch@ck: macro added . . . . .                                                            | 19                                                                                           |
| \@extrafloats: macro added . . . . .                                                      | 20                                                                                           |
| \newlanguage: New engine-specific<br>allocation scheme (latexrelease) . . . . .           | 17                                                                                           |
| 2014-12-30 ltspace.dtx v1.3b                                                              | 2015-01-11 lcounts.dtx v1.1h                                                                 |
| \@: \@ discards spaces when moving<br>(pr3039)(latexrelease) . . . . .                    | \TextOrMath: Add command to solve<br>robustness issues (pr/3752)<br>(latexrelease) . . . . . |
| 438                                                                                       | 520                                                                                          |
| 2015-01-03 ltdirchk.dtx v1.4a                                                             | 2015-01-11 lffloat.dtx v1.2b                                                                 |
| \typein: use modified definition in<br>luatex . . . . .                                   | \@dblfloatplacement: float order in<br>2-column (latexrelease) . . . . .                     |
| 82                                                                                        | 921                                                                                          |
| 2015-01-03 ltdirchk.dtx v1.1                                                              | \@xfloat: Check for valid option<br>(latexrelease) . . . . .                                 |
| General: Enable extra primitives when<br>LuaTeX is used . . . . .                         | 915                                                                                          |
| 3                                                                                         | \end@dblfloat: float order in<br>2-column (latexrelease) . . . . .                           |
| 2015-01-03 ltfinal.dtx v2.0a                                                              | 919                                                                                          |
| General: Skip resetting codes with<br>Unicode engines . . . . .                           | 2015-01-11 ltfssbas.dtx v3.0y                                                                |
| 1187                                                                                      | \@DeclarMathSizes: Allow arbitrary<br>units (latexrelease) . . . . .                         |
| Unicode data loading added . . . . .                                                      | 530                                                                                          |
| 1178                                                                                      | 2015-01-11 ltspace.dtx v1.3d                                                                 |
| 2015-01-07 ltvers.dtx v1.0n                                                               | \@Eshack: Allow hyphenation<br>(Donald Arseneau pr/3498)<br>(latexrelease) . . . . .         |
| \@check@IncludeInRelease: macro<br>added . . . . .                                        | 430                                                                                          |
| 39                                                                                        | \@esphack: Allow hyphenation<br>(Donald Arseneau pr/3498)<br>(latexrelease) . . . . .        |
| 2015-01-08 ltboxes.dtx v1.1h                                                              | 429                                                                                          |
| \framebox: Make Robust<br>(latexrelease) . . . . .                                        | 2015-01-14 ltoutput.dtx v1.2n                                                                |
| 826                                                                                       | \@addtocurcol: float order in<br>2-column (latexrelease) . . . . .                           |
| \makebox: Make Robust (latexrelease)                                                      | 1127                                                                                         |
| 819                                                                                       | \@addtobdblcol: float order in<br>2-column (latexrelease) . . . . .                          |
| \parbox: Make Robust (latexrelease)                                                       | 1137                                                                                         |
| 828                                                                                       | \@adtonextcol: float order in<br>2-column (latexrelease) . . . . .                           |
| \raisebox: Make Robust<br>(latexrelease) . . . . .                                        | 1134                                                                                         |
| 834                                                                                       | \@doclearpage: Empty kludgeins box<br>if necessary, pr/3528 . . . . .                        |
| \rule: Make Robust (latexrelease) . . . . .                                               | 1108                                                                                         |
| 833                                                                                       | float order in 2-column<br>(latexrelease) . . . . .                                          |
| \savebox: Make Robust (latexrelease)                                                      | 1108                                                                                         |
| 823                                                                                       | \@startdblcolumn: float order in<br>2-column (latexrelease) . . . . .                        |
| 2015-01-08 ltdefns.dtx v1.4a                                                              | 1121                                                                                         |
| \MakeRobust: Added macro . . . . .                                                        | \@xtryfc: float order in 2-column<br>(latexrelease) . . . . .                                |
| 92                                                                                        | 1123                                                                                         |
| 2015-01-08 ltlenth.dtx v1.1c                                                              | \@ztryfc: float order in 2-column<br>(latexrelease) . . . . .                                |
| \setlength: to ensure first length<br>argument is terminated.<br>(latexrelease) . . . . . | 1124                                                                                         |
| 521                                                                                       | 2015-01-14 ltspace.dtx v1.3e                                                                 |
| 2015-01-08 ltmath.dtx v1.1h                                                               | \@addpenalty: Avoid adding redundant<br>skips (DPC) . . . . .                                |
| \@: Make Robust (latexrelease) . . . . .                                                  | 433                                                                                          |
| \]: Make Robust (latexrelease) . . . . .                                                  | 2015-01-17 ltvers.dtx v1.0m                                                                  |
| 793                                                                                       | \@check@IncludeInRelease: modified<br>with \@currname . . . . .                              |
| 793                                                                                       | 39                                                                                           |
| 2015-01-09 ltfssini.dtx v3.1a                                                             | 2015-01-19 ltvers.dtx v1.0o                                                                  |
| \em: Allow \emph to produce small<br>caps (latexrelease) . . . . .                        | \@check@IncludeInRelease: Optional<br>argument . . . . .                                     |
| 654                                                                                       | 39                                                                                           |
| \emminershape: macro added<br>(latexrelease) . . . . .                                    |                                                                                              |
| 654                                                                                       |                                                                                              |
| 2015-01-09 ltspace.dtx v1.1h                                                              |                                                                                              |
| \addpenalty: Donald Arseneau's fix<br>from PR/377703 (latexrelease) . . . . .             |                                                                                              |
| 433                                                                                       |                                                                                              |

|                                                                                                                                                                 |                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2015-01-20 ltoutput.dtx v1.2m<br>\f@tracemessage: Reset<br>\IncludeInRelease flags . . . . . 1147                                                               | 2015-02-21 ltpictur.dtx v1.1k<br>General: Removed autoload code . . . . . 862                                                                                              |
| 2015-01-22 ltvers.dtx v1.0p<br>General: Preserve any \everyjob<br>material inserted by a loader (.ini<br>file) . . . . . 38                                     | 2015-02-21 lplain.dtx v2.0e<br>General: Removed autoload code . . . . . 14                                                                                                 |
| 2015-01-23 ltfinal.dtx v2.0b<br>\newmarks: use reserved count 256 1174<br>\newXeTeXintercharclass: use<br>reserved count 257 . . . . . 1175                     | 2015-02-21 ltab.dtx v1.1n<br>General: Removed autoload code . . . . . 837                                                                                                  |
| 2015-01-23 lplain.dtx v2.0c<br>\xtrafloats: reserve counts 256–265 20                                                                                           | 2015-02-21 ltvers.dtx v1.0r<br>General: Removed autoload code . . . . . 37                                                                                                 |
| 2015-01-24 ltfinal.dtx v2.0c<br>General: Skip T1-code entirely with<br>Unicode engines . . . . . 1178                                                           | 2015-02-21 ltvers.dtx v1.0w<br>\@check@IncludeInRelease: set<br>\@currname empty here (in case<br>\IncludeInRelease input early) . . . . . 39                              |
| 2015-02-03 ltfinal.dtx v2.0d<br>General: Set \lccode for - with<br>Unicode engines . . . . . 1179                                                               | 2015-02-22 ltfsscmp.dtx v3.0e<br>General: Moved all code into<br>latexrelease - obsolete commands<br>are no longer automatically part of<br>the kernel . . . . . 597       |
| 2015-02-16 ltoutenc.dtx v1.99m<br>General: Added \textcommabelow<br>latex/4414 . . . . . 481                                                                    | 2015-03-02 lplain.dtx v2.0f<br>\@mathgroup@top: macro added . . . . . 19<br>\newlanguage: allow 255 math groups<br>in Unicode engines . . . . . 17                         |
| 2015-02-16 ltoutenc.dtx v1.99n<br>General: Added \textcommaabove . . . . . 482<br>Added composites for ç . . . . . 490<br>Added composites for \c . . . . . 484 | 2015-03-10 lplain.dtx v2.0g<br>\hideoutput: macro added . . . . . 36<br>\loggingall: Reorganize to be less<br>noisy . . . . . 33<br>\tracingnone: macro added . . . . . 34 |
| 2015-02-16 lttextcomp.dtx v1.99m<br>General: Added lmtt (Heiko Oberdiek)<br>latex/4415 . . . . . 726                                                            | 2015-03-12 ltoutput.dtx v1.2m<br>General: initialise \@dbldeflist<br>again . . . . . 1098                                                                                  |
| 2015-02-19 ltvers.dtx v1.0q<br>\@check@IncludeInRelease: Swap<br>argument order . . . . . 39                                                                    | 2015-03-18 ltfssdcl.dtx v3.0q<br>\DeclareSymbolFont: Restrict Symbol<br>fonts to 0–15 . . . . . 615                                                                        |
| 2015-02-20 lplain.dtx v2.0d<br>\loggingall: Spell commands<br>correctly :-). . . . . 33                                                                         | \document@select@group: Introduce<br>\@mathgroup@top . . . . . 607                                                                                                         |
| 2015-02-21 ldefsns.dtx v1.4b<br>General: Removed autoload support . . . . . 80                                                                                  | \select@group: Introduce<br>\@mathgroup@top . . . . . 605                                                                                                                  |
| 2015-02-21 lterror.dtx v1.2o<br>General: Removed autoload support . . . . . 388                                                                                 | 2015-03-26 ltfinal.dtx v2.0d<br>General: Use renamed<br>unicode-letters.def . . . . . 1178                                                                                 |
| 2015-02-21 ltfiles.dtx v1.1m<br>General: Removed autoload support . . . . . 442                                                                                 | 2015-04-07 ltfssbas.dtx v3.1a<br>\wrong@fontshape: Try loading fd file<br>if family has changed . . . . . 542                                                              |
| 2015-02-21 ltfssbas.dtx v3.0z<br>General: Removed autoload code . . . . . 523                                                                                   | 2015-04-28 ltfinal.dtx v2.0f<br>\newXeTeXintercharclass: define<br>\@alloc@intercharclass for<br>compatibility with older xelatex<br>initialization . . . . . 1175         |
| 2015-02-21 ltfsscmp.dtx v3.0d<br>General: Removed autoload code . . . . . 597                                                                                   | 2015-05-10 ltlists.dtx v1.0t<br>\@doendpe: Explicitly reset<br>\clubpenalty before clearing<br>\everypar; see also pr/0462 and<br>pr/4065 . . . . . 812                    |
| 2015-02-21 ltfssdcl.dtx v3.0p<br>General: Removed autoload code . . . . . 602                                                                                   |                                                                                                                                                                            |
| 2015-02-21 lfsstrc.dtx v3.0k<br>General: Removed autoload code . . . . . 572                                                                                    |                                                                                                                                                                            |
| 2015-02-21 ltoutenc.dtx v1.99m<br>General: Removed autoload code . . . . . 466                                                                                  |                                                                                                                                                                            |
| 2015-02-21 ltoutput.dtx v1.2n<br>General: Removed autoload code . . . . . 1087                                                                                  |                                                                                                                                                                            |
| \f@depth: macro added(latexrelease) 1107                                                                                                                        |                                                                                                                                                                            |

|                                                                                                            |                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|-----|
| 2015-06-19 ltfinal.dtx v2.0g                                                                               | \newluafunction: Macro added . . . . .                                                                     | 51  |
| \@alloc@intercharclass@top: Use<br>–1 for first range to get contiguous<br>allocation . . . . .            | 1175                                                                                                       |     |
| \newmarks: Use –1 for first range to<br>get contiguous allocation . . . . .                                | 1174                                                                                                       |     |
| 2015-06-19 ltplain.dtx v2.0h                                                                               | \module_error: Function added . . . . .                                                                    | 56  |
| General: delete spurious old definition<br>of \newtoks . . . . .                                           | 23                                                                                                         |     |
| \@alloc: extra braces in case<br>arguments not single token . . . . .                                      | 19                                                                                                         |     |
| \newlanguage: Use –1 for first range<br>to get contiguous allocation . . . . .                             | 17                                                                                                         |     |
| 2015-06-23 ltfinal.dtx v2.0h                                                                               | \module_info: Function added . . . . .                                                                     | 56  |
| General: set \patch@level in ltvers<br>rather than in ltfinal/ltpatch . . . . .                            | 1190                                                                                                       |     |
| 2015-06-23 ltvers.dtx v1.0t                                                                                | \module_warning: Function added . . . . .                                                                  | 56  |
| General: set \patch@level in ltvers<br>rather than in ltfinal/ltpatch . . . . .                            | 37                                                                                                         |     |
| 2015-08-06 ltplain.dtx v2.0i                                                                               | \modules: Function modified . . . . .                                                                      | 54  |
| \extrafloats: Add \string in case<br>argument is not an unexpandable<br>primitive . . . . .                | 20                                                                                                         |     |
| 2015-08-23 ltdirchk.dtx v1.2                                                                               | \create_callback: Function added . . . . .                                                                 | 67  |
| General: Do not use luatex prefix . . . . .                                                                | 3                                                                                                          |     |
| 2015-08-23 ltvers.dtx v1.0v                                                                                | \provides_module: Function added . . . . .                                                                 | 54  |
| General: Allow negative patchlevel for<br>pre-release . . . . .                                            | 38                                                                                                         |     |
| 2015-08-30 ltplain.dtx v2.1a                                                                               | \luatexbase: Table added . . . . .                                                                         | 54  |
| \newinsert: new \newinsert<br>implementation . . . . .                                                     | 22                                                                                                         |     |
| 2015-09-205 ltoutput.dtx v1.3a                                                                             | 2015-10-02 ltdirchk.dtx v1.2a                                                                              | 3   |
| General: extended \@freelist . . . . .                                                                     | General: Allow backing out of<br>unprefixed names . . . . .                                                | 53  |
| 2015-09-24 ltluatex.dtx v1.0a                                                                              | 2015-10-02 ltluatex.dtx v1.0b                                                                              | 53  |
| \call_callback: Function added . . . . .                                                                   | General: Fix backing out of TeX code . . . . .                                                             | 53  |
| \callback_register: Function<br>modified . . . . .                                                         | 2015-10-02 ltluatex.dtx v1.0c                                                                              | 53  |
| \callback_descriptions: Function<br>added . . . . .                                                        | General: Allow backing out of Lua<br>code . . . . .                                                        | 53  |
| \catcodetable@atletter: Macro<br>added . . . . .                                                           | 2015-10-02 ltluatex.dtx v1.0e                                                                              | 53  |
| \catcodetable@initex: Macro added . . . . .                                                                | \uninstall: Function added . . . . .                                                                       | 72  |
| \catcodetable@latex: Macro added . . . . .                                                                 | 2015-10-03 ltluatex.dtx v1.0f                                                                              | 53  |
| \catcodetable@string: Macro added . . . . .                                                                | \provides_module: use luatexbase_log . . . . .                                                             | 54  |
| \add_to_callback: Function added . . . . .                                                                 | 2015-10-27 ltplain.dtx v2.1b                                                                               | 53  |
| \remove_from_callback: Function<br>added . . . . .                                                         | \extrafloats: Use global assignment<br>when switching to extended range . . . . .                          | 20  |
| \new_attribute: Function added . . . . .                                                                   | 2015-11-07 ltspace.dtx v1.3f                                                                               | 53  |
| \disable_callback: Function added . . . . .                                                                | \@esphack: Only space if there is no<br>space at the end of the hlist<br>latex/4443 . . . . .              | 429 |
| \in_callback: Function added . . . . .                                                                     | 2015-11-14 ltluatex.dtx v1.0g                                                                              | 53  |
| \newlanguage: Extended stream<br>allocation in luatex (0.85) . . . . .                                     | General: Track LuaTeX changes for<br>(token).create . . . . .                                              | 56  |
| 2015-11-18 ltplain.dtx v2.2a                                                                               | 2015-11-18 ltplain.dtx v2.2a                                                                               | 53  |
| \newlanguage: Only extend allocation<br>of write streams (see luatex list) . . . . .                       | \newlanguage: Extended stream<br>allocation in luatex (0.85) . . . . .                                     | 17  |
| 2015-11-19 ltplain.dtx v2.2b                                                                               | 2015-11-19 ltplain.dtx v2.2b                                                                               | 53  |
| \newlanguage: Only extend allocation<br>of write streams (see luatex list) . . . . .                       | \newlanguage: Only extend allocation<br>of write streams (see luatex list) . . . . .                       | 17  |
| 2015-11-27 ltluatex.dtx v1.0h                                                                              | 2015-11-27 ltluatex.dtx v1.0h                                                                              | 53  |
| \callback_descriptions: Match test<br>in in-callback latex/4445 . . . . .                                  | \call_callback: Guard against undefined<br>list latex/4445 . . . . .                                       | 71  |
| \in_callback: Guard against undefined<br>list latex/4445 . . . . .                                         | 2015-11-29 ltluatex.dtx v1.0i                                                                              | 53  |
| \call_callback: Check name is not nil<br>in error message (PHG) . . . . .                                  | General: Declare this as local before<br>used in the module error<br>definitions (PHG) . . . . .           | 54  |
| \create_callback: Check name is not<br>nil in error message (PHG) . . . . .                                | \call_callback: Check name is not nil<br>in error message (PHG) . . . . .                                  | 67  |
| 2015-12-02 ltluatex.dtx v1.0j                                                                              | \create_callback: Check name is not<br>nil in error message (PHG) . . . . .                                | 67  |
| General: Adjust hashtokens to store<br>the result of tex.hashtokens(), not<br>the function (PHG) . . . . . | General: Adjust hashtokens to store<br>the result of tex.hashtokens(), not<br>the function (PHG) . . . . . | 56  |

|                                                                                                  |      |                                                                                          |
|--------------------------------------------------------------------------------------------------|------|------------------------------------------------------------------------------------------|
| Assorted typos fixed (PHG) . . . . .                                                             | 46   | 2016-02-18 ltfssdcl.dtx v3.0r                                                            |
| Declaration/use of first_ head fixed (PHG) . . . . .                                             | 55   | \@DeclareMathDelimiter: Check for delimiter not \delimiter . . . . . 626                 |
| Remove nonlocal iteration variables (PHG) . . . . .                                              | 46   | \@DeclareMathAccent: Check for mathaccent not \mathaccemt . . . . . 622                  |
| Remove unreachable code after calls to error() (PHG) . . . . .                                   | 46   | \@DeclareMathRadical: Check for radical not \radical . . . . . 629                       |
| 2015-12-02 ltluatex.dtx v1.0k                                                                    |      | \@DeclareMathSymbol: Check for mathchar not \mathchar . . . . . 624                      |
| General: resolve name and i.description (PHG) . . . . .                                          | 65   | 2016-03-13 ltluatex.dtx v1.0n                                                            |
| call_callback: Give more specific error messages (PHG) . . . . .                                 | 67   | General: contribute_ filter added . . . . . 63                                           |
| add_to_callback: Give more specific error messages (PHG) . . . . .                               | 68   | insert_ local_ par added . . . . . 63                                                    |
| remove_from_callback: adjust initialization of cb local (PHG) . . . . .                          | 70   | 2016-03-29 ltpictur.dtx v1.11                                                            |
| Give more specific error messages (PHG) . . . . .                                                | 70   | \@oval: add setting of line tests 884, 885                                               |
| create_callback: Give more specific error messages (PHG) . . . . .                               | 67   | initialise tests . . . . . 884                                                           |
| 2015-12-10 ltfinal.dtx v2.0i                                                                     |      | \@ovhorz: use glue not leaders if horizontal line not required . . . . . 887             |
| General: Use new common Unicode data loaders . . . . .                                           | 1178 | \@ovvert: use glue not leaders if vertical line not required . . . . . 886               |
| 2015-12-18 ltluatex.dtx v1.0l                                                                    |      | \if@ovhline: macro added (latex/4452) . . . . . 884                                      |
| General: Load Unicode data from source . . . . .                                                 | 49   | \if@ovvline: macro added (latex/4452) . . . . . 884                                      |
| 2016-01-04 ltfinal.dtx v2.0j                                                                     |      | 2016-04-22 ltfinal.dtx v2.0q                                                             |
| General: Do not set up inter character classes for XeTeX . . . . .                               | 1178 | \@e@alloc@intercharclass@top: XeTeX 0.99996 has 4096 char classes not 256 . . . . . 1175 |
| \@e@alloc@intercharclass@top: Start allocation at one not three . . . . .                        | 1175 | 2016-06-19 ltoutenc.dtx v1.99m                                                           |
| 2016-01-05 ltfinal.dtx v2.0k                                                                     |      | General: OT1 definition (was duplicate T1 definition) . . . . . 484                      |
| \@e@alloc@intercharclass@top: Remove duplicated code . . . . .                                   | 1175 | 2016-06-20 ltclass.dtx v1.1j                                                             |
| 2016-01-05 ltfinal.dtx v2.0l                                                                     |      | General: don't declare as \onlypreamble . . . . . 984                                    |
| General: Correct latexrelease guards . . . . .                                                   | 1178 | 2016-07-29 lplain.dtx v2.2c                                                              |
| Ensure old definitions for inter-character class toks are available using latexrelease . . . . . | 1178 | \extrafloats: use \global \chardef 20 \newinsert: fix for tlb-newinsert-001 . . . . . 22 |
| Missing brace . . . . .                                                                          | 1178 | 2016-10-02 ltclass.dtx v1.2a                                                             |
| 2016-01-05 ltfinal.dtx v2.0m                                                                     |      | \@ifclasswith: Ignore spaces while checking for option clash . . . . . 985               |
| General: Undefine XeTeX classes when using patching an older kernel . . . . .                    | 1178 | \ExecuteOptions: Ignore spaces in argument . . . . . 995                                 |
| 2016-01-05 ltfinal.dtx v2.0p                                                                     |      | 2016-10-15 ldirchk.dtx v1.2b                                                             |
| General: Only apply XeTeX change if XeTeX is in use . . . . .                                    | 1178 | General: Require eT <sub>EX</sub> . . . . . 3                                            |
| 2016-02-11 ltluatex.dtx v1.0m                                                                    |      | 2016-10-15 lterror.dtx v1.2p                                                             |
| General: pdf_stream_filter_callback removed . . . . .                                            | 64   | General: Require eT <sub>EX</sub> . . . . . 388                                          |
| process_rule, [hv]pack_quality append_to_vlist_filter added . . . . .                            | 63   | 2016-10-15 ltfinal.dtx v2.0r                                                             |
| read_cidmap_file added . . . . .                                                                 | 62   | General: Require eT <sub>EX</sub> . . . . . 1174                                         |
| show_warning_message added . . . . .                                                             | 63   | 2016-10-15 lplain.dtx v2.2d                                                              |
| token_filter removed . . . . .                                                                   | 63   | General: Require eT <sub>EX</sub> . . . . . 14                                           |

|                                                                                          |      |                                                                                                     |
|------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------|
| 2016-10-16 ltplain.dtx v2.3a                                                             |      | 2017-02-19 ltoutenc.dtx v2.0f                                                                       |
| \newlanguage: Allow languages up to<br>16383 in luatex . . . . .                         | 17   | General: add \empty to guard against<br>3rd argument being empty . . . . .                          |
| 2016-10-19 lccounts.dtx v1.1j                                                            |      | 483                                                                                                 |
| \TextOrMath: Test directly for<br>\protected . . . . .                                   | 520  | declare composites with empty base<br>for hat and tilde, use same slots for<br>\textasciicircum ans |
| 2016-11-06 ltplain.dtx v2.3b                                                             |      | 498                                                                                                 |
| General: Drop \outer entirely . . . . .                                                  | 14   | \textasciitilde . . . . .                                                                           |
| 2016-11-09 ltclass.dtx v2.1b                                                             |      | 498                                                                                                 |
| \@fileswithoptions: Improve \ifx<br>tests PR/4497 . . . . .                              | 999  | declare straight quotes using new<br>\remove@tlig command . . . . .                                 |
| 2016-11-17 ltluatex.dtx v1.0p                                                            |      | 2017-02-22 ltoutenc.dtx v2.0g                                                                       |
| General: call_edit added . . . . .                                                       | 63   | General: Fix typo introduced at 2.0f . . . . .                                                      |
| 2016-12-03 fontdef.dtx v3.0a                                                             |      | 2017-02-24 ltoutenc.dtx v2.0h                                                                       |
| General: (DPC) Default to TU<br>encoding for Unicode TeX engines . . . . .               | 664  | General: introduce                                                                                  |
| \shapedefault: (DPC) Default to TU<br>encoding for Unicode TeX engines . . . . .         | 668  | \DeclareUnicodeAccent . . . . .                                                                     |
| 2016-12-04 ltoutenc.dtx v2.0a                                                            |      | \DeclareTextCompositeCommand: add                                                                   |
| General: Added TU encoding . . . . .                                                     | 498  | check whether the accent command<br>is defined for this encoding . . . . .                          |
| 2017-01-01 ltoutput.dtx v1.3b                                                            |      | 473                                                                                                 |
| General: make fpmin negative so<br>ignored even if float height is<br>negative . . . . . | 1155 | 2017-03-08 ltclass.dtx v1.2c                                                                        |
| 2017-01-10 ltfsbas.dtx v3.2a                                                             |      | General: add \@parse@version@dash<br>to support yyyy-mm-dd as well as<br>yyyy/mm/dd . . . . .       |
| \showhyphens: Add version of<br>\showhyphens that works with<br>XeTeX . . . . .          | 547  | 984                                                                                                 |
| 2017-01-23 ltoutenc.dtx v2.0b                                                            |      | 2017-03-09 ltfinal.dtx v2.0t                                                                        |
| General: Added TU specific commands<br>in ASCII range pr/4500 . . . . .                  | 498  | \l@nohyphenation: ensure<br>\l@nohyphenation is defined. . . . .                                    |
| 2017-01-24 ltoutenc.dtx v2.0c                                                            |      | 1181                                                                                                |
| General: Declare TU composites for i<br>and j . . . . .                                  | 498  | 2017-03-09 ltmiscen.dtx v1.1m                                                                       |
| Make \textasteriskcentered<br>U+2217 not U+204E . . . . .                                | 498  | \overline: Use \language not<br>\hyphenchar . . . . .                                               |
| TeX ligature syntax for xetex and<br>luatex reversed . . . . .                           | 498  | 778                                                                                                 |
| 2017-01-24 ltoutenc.dtx v2.0d                                                            |      | \verb: Use \language to stop<br>hyphenation . . . . .                                               |
| General: Declare macron composites<br>for YyGg . . . . .                                 | 498  | 783                                                                                                 |
| 2017-02-12 ltoutenc.dtx v2.0e                                                            |      | 2017-03-10 ltfiles.dtx v1.1n                                                                        |
| General: Declare fallback code for<br>\textasteriskcentered . . . . .                    | 498  | \document: Save language default . . . . .                                                          |
| 2017-02-18 ltluatex.dtx v1.1c                                                            |      | 445                                                                                                 |
| new_attribute: Parameterize count<br>used in tracking . . . . .                          | 57   | 2017-03-10 ltoutput.dtx v1.3c                                                                       |
| new_bytocode: Parameterize count<br>used in tracking . . . . .                           | 58   | \@writesetup: Reset \language . . . . .                                                             |
| new_chunkname: Parameterize count<br>used in tracking . . . . .                          | 58   | 1115                                                                                                |
| new_whatsit: Parameterize count used<br>in tracking . . . . .                            | 58   | 2017-03-13 ltdefns.dtx v1.5a                                                                        |
|                                                                                          |      | \l@-: Define \l@- in terms of \hyphenchar . . . . .                                                 |
|                                                                                          |      | 110                                                                                                 |
|                                                                                          |      | 2017-03-27 ltdefns.dtx v1.5b                                                                        |
|                                                                                          |      | \@dischyp: Define \@dischyp after<br>\l@- . . . . .                                                 |
|                                                                                          |      | 110                                                                                                 |
|                                                                                          |      | 2017-03-28 ltluatex.dtx v1.1e                                                                       |
|                                                                                          |      | General: glyph_stream_provider<br>added . . . . .                                                   |
|                                                                                          |      | 63                                                                                                  |
|                                                                                          |      | 2017-03-29 ltboxes.dtx v1.3a                                                                        |
|                                                                                          |      | \arrayparboxrestore: Reset<br>\lineskiplimit . . . . .                                              |
|                                                                                          |      | 830                                                                                                 |
|                                                                                          |      | 2017-04-05 ltoutenc.dtx v2.0i                                                                       |
|                                                                                          |      | \DeclareTextCompositeCommand:                                                                       |
|                                                                                          |      | Declare accent command if not<br>already declared when declaring a<br>composite. . . . .            |
|                                                                                          |      | 473                                                                                                 |
|                                                                                          |      | 2017-04-10 ltplain.dtx v2.3c                                                                        |
|                                                                                          |      | \newlanguage: Correction to code to<br>skip write18 in luatex . . . . .                             |
|                                                                                          |      | 17                                                                                                  |

|                                                                                                                           |      |                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2017-04-11 ltoutput.dtx v2.4a                                                                                             |      | 2018-05-08 ltclass.dtx v1.2i                                                                                                                                 |
| \newpage: account for the depth of the<br>last row of the page . . . . .                                                  | 1101 | \pkgcls@parse@date@arg: Make<br>suspicious rollback a warning not<br>error: github issue 43 . . . . .                                                        |
| 2017-12-17 ltoutput.dtx v1.4b                                                                                             |      | 1021                                                                                                                                                         |
| \@addtonextcol: fix doc guards . . .                                                                                      | 1134 | 2018-05-11 ltfinal.dtx v2.18                                                                                                                                 |
| 2018-01-06 ltdefns.dtx 1.5c                                                                                               |      | General: Make invalid UTF-8 also safe,<br>for legacy filesystem encodings . . . . .                                                                          |
| \@ifundefined: Avoid defining<br>undefined commands to \relax . . .                                                       | 106  | 1184                                                                                                                                                         |
| 2018-02-18 ltclass.dtx v1.2d                                                                                              |      | 2018-05-29 ltclass.dtx v1.2j                                                                                                                                 |
| \@ifl@ter: Added 0 up front to make<br>bad data come out as 0. . . . .                                                    | 984  | \endfilecontents: use \csname not<br>\@undefined . . . . .                                                                                                   |
| General: Introduce rollback concept                                                                                       | 1018 | 1013                                                                                                                                                         |
| 2018-03-08 ltcnts.dtx v1.1k                                                                                               |      | 2018-08-11 ltoutenc.dtx v2.0j                                                                                                                                |
| \@ifbothcounters: Interface added . .                                                                                     | 516  | General: Provide \guillemetleft and<br>\guillemetright . . . . .                                                                                             |
| \@removefromreset: Interface added . .                                                                                    | 515  | 486, 492, 501                                                                                                                                                |
| \counterwithin: Interface added . . .                                                                                     | 517  | 2018-08-18 ltluatex.dtx v1.1h                                                                                                                                |
| 2018-03-24 ltclass.dtx v1.2e                                                                                              |      | General: append_to_vlist_ filter is<br>exclusive . . . . .                                                                                                   |
| \pkgcls@use@this@release: Use full<br>file name for old release . . . . .                                                 | 1023 | 63                                                                                                                                                           |
| 2018-03-25 ltfinal.dtx v2.1a                                                                                              |      | 2018-08-24 ltfinal.dtx v2.1f                                                                                                                                 |
| General: default to UTF-8 . . . . .                                                                                       | 1182 | \document@default@language: Add to<br>latexrelease (github/68) . . . . .                                                                                     |
| \UseRawInputEncoding: Macro<br>added . . . . .                                                                            | 1183 | 1181                                                                                                                                                         |
| 2018-03-27 ltclass.dtx v1.2f                                                                                              |      | 2018-09-02 ltsect.dtx v1.1b                                                                                                                                  |
| \endfilecontents: Use full file name<br>for old release . . . . .                                                         | 1012 | \@dottedtocline: Prevent protrusion<br>( <a href="https://tex.stackexchange.com/q/172785/10109">https://tex.stackexchange.com/q/172785/10109</a> ) . . . . . |
| 2018-04-06 ltfinal.dtx v2.1b                                                                                              |      | 909                                                                                                                                                          |
| \UseRawInputEncoding: Undo changes<br>to \DeclareFontEncoding@ and<br>definition of<br>\DeclareUnicodeCharacter . . . . . | 1183 | 2018-09-24 fontdef.dtx v3.0b                                                                                                                                 |
| 2018-04-07 ltfinal.dtx v2.1c                                                                                              |      | General: Start LR-mode if necessary<br>(git/49) . . . . .                                                                                                    |
| \UseRawInputEncoding: Undefine<br>\inputencodingname . . . . .                                                            | 1183 | 680                                                                                                                                                          |
| 2018-04-08 ltclass.dtx v1.2g                                                                                              |      | 2018-09-24 ltmath.dtx v1.2b                                                                                                                                  |
| \@ifl@ter: Strip leading spaces from<br>dates. . . . .                                                                    | 984  | \smash: Start LR-mode if necessary<br>(git/49) . . . . .                                                                                                     |
| 2018-04-08 ltclass.dtx v1.2h                                                                                              |      | 789                                                                                                                                                          |
| \onefilewithoptions: Pass<br>expanded date . . . . .                                                                      | 1019 | \phantom: Start LR-mode if<br>necessary (git/49) . . . . .                                                                                                   |
| 2018-04-08 ltfinal.dtx v2.1d                                                                                              |      | 788                                                                                                                                                          |
| General: Delay full UTF-8 handling to<br>\everyjob . . . . .                                                              | 1184 | 2018-09-24 ltspace.dtx v1.3h                                                                                                                                 |
| 2018-04-11 ltcnts.dtx v1.1l                                                                                               |      | \enspace: Start LR-mode if necessary<br>(git/49) . . . . .                                                                                                   |
| \counterwithin: Correct default<br>(issue/38) . . . . .                                                                   | 517  | 440                                                                                                                                                          |
| 2018-05-02 ltluatex.dtx v1.1g                                                                                             |      | \leavevmode@ifvmode: Macro added<br>(git/49) . . . . .                                                                                                       |
| General: find_sfd_file removed . . .                                                                                      | 62   | 440                                                                                                                                                          |
| finish_synctex_callback added . . .                                                                                       | 63   | 2018-09-26 ltdefns.dtx v1.5e                                                                                                                                 |
| glyph_not_found added . . . . .                                                                                           | 63   | \renew@command: Always explicitly<br>generate a space after the csname<br>and not rely on \noexpand to save<br>tokens (git/41) . . . . .                     |
| read_sfd_file removed . . . . .                                                                                           | 62   | 86                                                                                                                                                           |
|                                                                                                                           |      | 2018-09-26 ltmiscen.dtx v1.1n                                                                                                                                |
|                                                                                                                           |      | \writefile: Sometimes mask the<br>endline char when writing to files<br>(github/73) . . . . .                                                                |
|                                                                                                                           |      | 767                                                                                                                                                          |
|                                                                                                                           |      | \add@percent@to@temptokena:<br>Sometimes mask the endline char<br>when writing to files (github/73) . . . . .                                                |
|                                                                                                                           |      | 767                                                                                                                                                          |
|                                                                                                                           |      | \protected@file@percent: Sometimes<br>mask the endline char when<br>writing to files (github/73) . . . . .                                                   |
|                                                                                                                           |      | 766                                                                                                                                                          |
|                                                                                                                           |      | 2018-09-26 ltsect.dtx v1.1c                                                                                                                                  |
|                                                                                                                           |      | \addcontentsline: Sometimes mask<br>the endline char when writing to                                                                                         |

|                                                                                                              |      |                                                                                                   |
|--------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------|
| files (github/73) . . . . .                                                                                  | 907  | 2019-02-07 ltfssbas.dtx v3.2b                                                                     |
| 2018-10-10 ltspace.dtx v1.3i                                                                                 |      | \define@newfont: Changed wording of warning (github/107) . . . . . 537                            |
| \@esphack: Don't introduce breakpoints if @nobreak is true and after sections . . . . .                      | 429  |                                                                                                   |
| 2018-10-11 ltmissen.dtx v1.1o                                                                                |      | 2019-06-18 lthuataex.dtx v1.1j                                                                    |
| \@osverb: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .                    | 781  | General: finish_synthex_callback renamed finish_synthex . . . . . 63                              |
| \@setupverbvisibleinspace: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .   | 780  | font_descriptor_objnum_provider added . . . . . 63                                                |
| \@verbvisiblespacebox: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .       | 779  | make_extensible added . . . . . 63                                                                |
| \@asciispace: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .                | 779  | new_graf added . . . . . 63                                                                       |
| \verbatim*: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .                  | 779  | page_objnum_provider added . . . . . 63                                                           |
| \verbvisibleinspace: Provide \verbvisibleinspace such that it is usable in normal text (github/70) . . . . . | 779  | process_pdf_image_content added . . . . . 63                                                      |
| Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) . . . . .                              | 779  | wrapup_run added . . . . . 63                                                                     |
| 2018-10-21 lthuataex.dtx v1.1i                                                                               |      | 2019-07-01 ltclass.dtx v1.3a                                                                      |
| new_luafunction: Function added . . . . .                                                                    | 59   | \endfilecontents: Support UTF8 and spaces in filecontents environment file name . . . . . 1010    |
| 2018-11-09 ltbibl.dtx LaTeXe2                                                                                |      | 2019-07-01 ltfssbas.dtx v1.2a                                                                     |
| \bibliography: Zap spaces in the argument as BibTeX doesn't support them (github/88) . . . . .               | 939  | \IfFileExists: Support UTF-8 . . . . . 456                                                        |
| 2018-11-18 ltoutenc.dtx v2.0k                                                                                |      | \Includeonly: Support UTF-8 . . . . . 450                                                         |
| General: Provide \Hwithstroke and \hwithstroke . . . . .                                                     | 503  | \set@curr@file: Support UTF-8 . . . . . 455                                                       |
| 2018-11-19 ltoutenc.dtx v2.0k                                                                                |      | 2019-07-09 ltfssbas.dtx v3.2c                                                                     |
| General: Added \Hwithstroke and \hwithstroke . . . . .                                                       | 485  | \DeclareErrorFont: Don't set any \f@... macros . . . . . 541                                      |
| 2018-11-28 ltoutput.dtx v1.4d                                                                                |      | 2019-07-09 ltfssini.dtx v3.1c                                                                     |
| \@combinedblfloats: Unbox \@outputbox to preserve boxing level (github/94) . . . . .                         | 1120 | General: Explicitly set some defaults . . . . . 659                                               |
| 2018-12-30 lttab.dtx v1.1p                                                                                   |      | 2019-08-22 ltxref.dtx v1.1l                                                                       |
| \@tabclassz: Add extra \hskip to guard against an \unskip at the start of a c-column cell (gh/102) . . . . . | 857  | \labelformat: Commanded moved from variorref.sty . . . . . 744                                    |
| 2019-02-07 ltfilehook.dtx v1.1o                                                                              |      | \Ref: Commanded moved from variorref.sty . . . . . 744                                            |
| \unqu@tefilef@und: Expand \@filef@und before executing second argument (github/109) . . . . .                | 1046 | \refstepcounter: Allow \p@... to have an argument . . . . . 743                                   |
| 2019-02-07 ltfssbas.dtx v1.1o                                                                                |      | 2019-08-27 fontdef.dtx v3.0c                                                                      |
| \@swaptwoargs: Helper macro added . . . . .                                                                  | 460  | General: Various commands made robust throughout the file . . . . . 673                           |
|                                                                                                              |      | 2019-08-27 ltboxes.dtx v1.3b                                                                      |
|                                                                                                              |      | General: Various commands made robust . . . . . 818                                               |
|                                                                                                              |      | 2019-08-27 ltclass.dtx v1.3b                                                                      |
|                                                                                                              |      | \endfilecontents: Make various commands robust . . . . . 1010                                     |
|                                                                                                              |      | 2019-08-27 ltdefns.dtx v1.5f                                                                      |
|                                                                                                              |      | General: Make various commands robust . . . . . 111                                               |
|                                                                                                              |      | \MakeRobust: Make the assignments global as we may need to apply them inside a group . . . . . 92 |
|                                                                                                              |      | 2019-08-27 ltfilehook.dtx v1.2b                                                                   |
|                                                                                                              |      | \unqu@tefilef@und: Make command robust . . . . . 1046                                             |

|                                                                     |      |                                                                                                     |          |
|---------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------|----------|
| 2019-08-27 ltfiles.dtx v1.2b                                        |      | 2019-08-30 lterror.dtx v1.2q                                                                        |          |
| \IfFileExists: Make command robust .....                            | 456  | \conditionally@traceoff: Macro added .....                                                          | 398      |
| 2019-08-27 ltfsbas.dtx v3.2d                                        |      | \conditionally@traceon: Macro added .....                                                           | 398      |
| General: Make various commands robust .....                         | 523  | 2019-09-09 ltfsdcl.dtx v3.0s                                                                        |          |
| \DeclareMathAccent: Make math accents robust .....                  | 622  | \DeclareMathSymbol: Allow definition if the math symbol was a command already robust .....          | 624      |
| \set@mathdelimiter: Make math delimiters robust .....               | 628  | 2019-09-11 ltclass.dtx v1.3c                                                                        |          |
| 2019-08-27 ltfsini.dtx v3.1d                                        |      | \endfilecontents: Support optional argument for filecontents .....                                  | 1010     |
| General: Make various commands robust .....                         | 634  | 2019-09-14 ltfinal.dtx v2.1h                                                                        |          |
| \@cucllist: Expand UTF8 chars when case changing (github/177) ..... | 1188 | 2019-09-16 ltxref.dtx v1.1m                                                                         |          |
| 2019-08-27 ltidxglo.dtx v1.1f                                       |      | General: Correctly revert the \p@... change .....                                                   | 744      |
| General: Make \index and \glossary robust .....                     | 934  | 2019-09-21 fontdef.dtx v3.0d                                                                        |          |
| 2019-08-27 ltlength.dtx v1.1d                                       |      | General: Distangle alias (gh/184) .....                                                             | 675, 679 |
| General: Make various command robust .....                          | 521  | 2019-10-02 ltexpl.dtx v0.0                                                                          |          |
| 2019-08-27 ltlogos.dtx v1.1j                                        |      | General: Initial version .....                                                                      | 73       |
| \TeX: Make \TeX command robust .....                                | 441  | 2019-10-02 ltfinal.dtx v2.2                                                                         |          |
| 2019-08-27 ltmath.dtx v1.2c                                         |      | General: Load ltexpl .....                                                                          | 1191     |
| General: Make various commands robust .....                         | 785  | 2019-10-02 ltluatex.dtx v1.1k                                                                       |          |
| 2019-08-27 ltmiscen.dtx v1.1p                                       |      | General: linebreak_ filter is exclusive .....                                                       | 63       |
| General: Make various commands robust .....                         | 760  | mlist_ to_ hlist is exclusive .....                                                                 | 63       |
| \begin: Make command robust .....                                   | 769  | process_ rule is exclusive .....                                                                    | 63       |
| \end: Make command robust .....                                     | 771  | 2019-10-07 ltab.dtx v1.1q                                                                           |          |
| 2019-08-27 ltoutput.dtx v1.4e                                       |      | \extracolsep: This needs to expand .....                                                            | 849      |
| \@begindvibox: Make \AtBeginDvi robust .....                        | 1099 | 2019-10-11 ltfiles.dtx v1.2c                                                                        |          |
| 2019-08-27 ltpage.dtx v1.0l                                         |      | \set@curr@file: Remove one brace group .....                                                        | 455      |
| General: Make various commands robust .....                         | 969  | 2019-10-11 ltfsstrc.dtx v3.0l                                                                       |          |
| 2019-08-27 ltpictur.dtx v1.1m                                       |      | \@font@aliasinfo: Added 'alias' size function .....                                                 | 595      |
| General: Make various commands robust .....                         | 862  | 2019-10-18 ltclass.dtx v1.3d                                                                        |          |
| Remove several unnecessary \gdef definitions .....                  | 862  | \load@onefilewithoptions: Initialize \...-h@k only when loading the package or class (gh/198) ..... | 1001     |
| 2019-08-27 ltsect.dtx v1.1d                                         |      | 2019-10-22 ltluatex.dtx v1.1j                                                                       |          |
| General: Make various commands robust .....                         | 898  | General: page_ objnum_ provider and process_ pdf_ image_ content classified data .....              | 63       |
| 2019-08-27 ltspace.dtx v1.3j                                        |      | 2019-10-25 ltmiscen.dtx v1.1q                                                                       |          |
| General: Make various commands robust .....                         | 421  | \add@percent@to@temptokena: Allow unbalanced conditionals in #1 (gh/202) .....                      | 767      |
| 2019-08-27 ltab.dtx v1.1q                                           |      | 2019-10-26 ltfiles.dtx v1.2d                                                                        |          |
| General: Make various commands robust .....                         | 837  | \@iffileonpath: quote on openin ...                                                                 | 459      |
| Remove several unnecessary \gdef definitions .....                  | 837  | \IfFileExists: don't quote name ..                                                                  | 456      |
|                                                                     |      | \IfFileExists@: quote on openin ...                                                                 | 457      |
|                                                                     |      | \set@curr@file: remove quotes .....                                                                 | 455      |

|                               |                                                                                                                         |                   |                                                                                                                               |      |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------|------|
| 2019-11-01 ltdirchk.dtx v1.3a | \filename@parse: take last . not first                                                                                  | 12                | \ttfamily: Streamlined implementation with hook                                                                               | 648  |
| 2019-11-02 ltmiscen.dtx v1.1s | \centercr: Make \centercr robust<br>(gh/203) . . . . .                                                                  | 774               | 2020-01-20 ltssdcl.dtx v3.0t                                                                                                  |      |
| 2019-11-02 ltspace.dtx v1.3k  | \normalcr: Make also \normalcr robust . . . . .                                                                         | 425               | \set@mathdelim: fix for gh/251                                                                                                | 628  |
| 2019-11-09 ltfiles.dtx v1.2e  | \set@curr@file: expand and \string before removing quotes . . . . .                                                     | 455               | 2020-01-20 ltoutenc.dtx v2.0n                                                                                                 |      |
| 2019-11-10 ltmiscen.dtx v1.1r | \add@percent@to@temptokena: fix to special comment catcodes<br>(gh/202) . . . . .                                       | 767               | General: fix for gh/251 . . . . .                                                                                             | 481  |
| 2019-11-11 ltfiles.dtx v1.2f  | \iffileonpath: make \file@und match quoting used on \openin .                                                           | 459               | 2020-01-22 lttextcomp.dtx v1.0b                                                                                               |      |
| 2019-11-22 ltoutenc.dtx v2.0l | General: Avoid spurious if fontenc selects LY1 as default encoding<br>(gh/199) . . . . .                                | 511               | \tc@subst: The overall default is \textcompsubstdefault not \substdefault . . . . .                                           | 698  |
| 2019-11-29 ltclass.dtx v1.3e  | \pr@videopackage: Protect package info text (gh/52) . . . . .                                                           | 988               | 2020-01-25 fontdef.dtx v3.0f                                                                                                  |      |
| 2019-12-17 fontdef.dtx v3.0e  | \mddefault: Set \bfdefault to "b"<br>\shadefont: Set \shadefont explicitly to "n"<br>\updefault: Set \updefault to "up" | 667<br>668<br>667 | General: Load t1enc.def last (gh/255)                                                                                         | 664  |
| 2019-12-17 ltnetcmd.dtx v3.4c | \textssc: Macro added . . . . .                                                                                         | 689               | 2020-01-25 ltoutenc.dtx v2.0m                                                                                                 |      |
| 2019-12-17 ltfsbas.dtx v3.2e  | \usefont: Don't call \fontseries or \fontshape . . . . .                                                                | 532               | General: Load each encoding file only once (gh/255) . . . . .                                                                 | 510  |
| 2019-12-17 ltfsini.dtx v3.1e  | General: Provide custom series settings a la mweights . . . . .                                                         | 635               | 2020-01-27 ltclass.dtx v1.3g                                                                                                  |      |
|                               | \DeclareEmphSequence: Provide \emph sequences . . . . .                                                                 | 653               | \endfilecontents: Fix typo in error message . . . . .                                                                         | 1012 |
| 2019-12-18 ltoutenc.dtx v2.0m | General: Don't fake \textcompwordmark; take default from T1 instead . . . . .                                           | 480               | 2020-01-28 ltclass.dtx v1.3h                                                                                                  |      |
|                               | \add@accent: Avoid code that breaks \accent . . . . .                                                                   | 473               | \endfilecontents: Allow spaces in option string and display only unknown options not the whole option list (gh/256) . . . . . | 1010 |
| 2019-12-21 fontdef.dtx v3.0e  | General: Distangle alias (gh/184)                                                                                       | 673–676           | 2020-01-31 ltvers.dtx v1.1e                                                                                                   |      |
| 2020-01-05 ltclass.dtx v1.3f  | \endfilecontents: Support more write streams in LuaTeX gh/238                                                           | 1010              | General: Allow for upcoming format as pre-release 0 . . . . .                                                                 | 38   |
| 2020-01-11 ltfsini.dtx v3.1f  | \rmfamily: Streamlined implementation with hook . . . . .                                                               | 648               | 2020-02-02 ltluatex.dtx v1.11                                                                                                 |      |
|                               | \ttfamily: Streamlined implementation with hook . . . . .                                                               | 648               | General: Add reverselist callback type glyph_info added . . . . .                                                             | 66   |
|                               |                                                                                                                         |                   | page_order_index added . . . . .                                                                                              | 63   |
|                               |                                                                                                                         |                   | post_linebreak_filter is reverselist . . . . .                                                                                | 63   |
|                               | \create_callback: Provide proper fallbacks for user-defined callbacks without user-provided default handler . . . . .   | 67                | \declareFontSeriesDefault:                                                                                                    |      |
|                               |                                                                                                                         |                   | Clarified error text . . . . .                                                                                                | 636  |
|                               |                                                                                                                         |                   | Corrected misspelled csname (gh/264) . . . . .                                                                                | 636  |
|                               | \symbol: XeTeX-specific version to avoid bug in maths mode. . . . .                                                     | 656               | 2020-02-05 ltoutenc.dtx v2.0n                                                                                                 |      |
|                               | \fontseries: Switch \if@forced@series added . . . . .                                                                   | 560               | General: Changed the package default to info (gh/262) . . . . .                                                               | 716  |
|                               | \fontseriesforce: Switch \if@forced@series added . . . . .                                                              | 560               | Ensure we are on a new format (gh/260) . . . . .                                                                              | 716  |
|                               | \symbol: XeTeX-specific version to avoid bug in maths mode. . . . .                                                     | 656               | 2020-02-07 ltfsini.dtx v3.1h                                                                                                  |      |
|                               | \fontseries: Switch \if@forced@series added . . . . .                                                                   | 560               | \symbol: XeTeX-specific version to avoid bug in maths mode. . . . .                                                           | 656  |
|                               | \fontseriesforce: Switch \if@forced@series added . . . . .                                                              | 560               | \fontseries: Switch \if@forced@series added . . . . .                                                                         | 560  |

|                                                                                                                                                      |      |                                                                                                                                                                     |          |
|------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| \if@forced@series: Switch<br>\if@forced@series added .....                                                                                           | 560  | 2020-02-25 ltfssini.dtx v3.1j<br>\bf@def@ult: Drop surplus “m” from<br>\bfdef@ult and \mddef@ult<br>(gh/291) .....                                                  | 646      |
| 2020-02-10 ltfssini.dtx v3.1h<br>\@defaultfamilyhook: Add<br>\@defaultfamilyhook to<br>\normalfont (gh/269) .....                                    | 649  | \prepare@family@series@update:<br>Drop surplus “m” from<br>\target@series@value (gh/291) .....                                                                      | 640      |
| \reset@font: Add<br>\@defaultfamilyhook to<br>\normalfont (gh/269) .....                                                                             | 657  | \update@series@target@value: Drop<br>surplus “m” from \reserved@d<br>(gh/291) .....                                                                                 | 640, 641 |
| 2020-02-10 lttextcomp.dtx v1.0c<br>General: Use \tabacckludge for<br>tabbing where necessary (gh/271) .....                                          | 701  | 2020-02-27 ltdefns.dtx v1.5g<br>\@gobblethree: Macro added .....                                                                                                    | 89       |
| 2020-02-11 fontdef.dtx v3.0g<br>General: Provide value for<br>\@fontenc@load@list (gh/273) .....                                                     | 664  | 2020-02-27 ltfssaxes.dtx v1.0d<br>\series@maybe@drop@one@m: Drop<br>“m” in certain values from a fixed<br>list (gh/293) .....                                       | 564      |
| 2020-02-11 ltfssini.dtx v3.1h<br>General: Provide default value for<br>\@fontenc@load@list (gh/273) .....                                            | 659  | \set@target@series: Drop “m” only<br>in a specific set of values (gh/293) .....                                                                                     | 563      |
| 2020-02-11 ltoutenc.dtx v2.0o<br>General: Update<br>\@fontenc@load@list with option<br>list (gh/273) .....                                           | 511  | 2020-02-27 ltfssbas.dtx v3.2g<br>\DeclareFontShape@: Only “m” if the<br>series value is a member of a fixed<br>list and issue warning if doing it<br>(gh/293) ..... | 524      |
| 2020-02-14 ltpictur.dtx v1.1n<br>\linethickness: Suppress spaces<br>following the declaration (gh/274) .....                                         | 867  | 2020-03-02 ltexpl.dtx v1.0a<br>General: Don’t load expl3 if already in<br>the format (gh/295) .....                                                                 | 73       |
| 2020-02-18 ltfssini.dtx v3.1i<br>\bfseries: Make the \ifx selection<br>outside of \fontseries argument<br>so that it is not done several times ..... | 643  | 2020-03-05 ltexpl.dtx v1.1<br>General: Load xparseltx if<br>\NewDocumentCommand is not<br>defined by expl3.ltx .....                                                | 73       |
| \mdseries: Make the \ifx selection<br>outside of \fontseries argument<br>so that it is not done several times .....                                  | 644  | 2020-03-06 ltboxes.dtx v1.3c<br>\clap: Macro \clap added .....                                                                                                      | 836      |
| \prepare@family@series@update: No<br>series auto-update when forced<br>(gh/277) .....                                                                | 639  | 2020-03-07 ltluatex.dtx v1.1m<br>remove_from_callback: Do not call<br>callback.register for user-defined<br>callbacks .....                                         | 70       |
| Recognize current family if it is not<br>a “meta” family and auto-update<br>series using \bfdefault (gh/277) .....                                   | 639  | 2020-03-07 ltmath.dtx v1.2e<br>\negthinspace: Add amsmath<br>math/text spacing commands to<br>the kernel (gh/303) .....                                             | 791      |
| 2020-02-18 ltmath.dtx v1.2d<br>\mathindent: Make \mathindent a<br>skip register to match amsmath<br>(gh/252) .....                                   | 798  | 2020-03-07 ltspace.dtx v1.3l<br>General: Moved \thinspace,<br>\negthinspace and \, to<br>ltmath.dtx (gh/303) .....                                                  | 421      |
| \equation: Separate formula and eqn<br>number by at least a space in fleqn<br>option .....                                                           | 799  | 2020-03-19 fontdef.dtx v3.0h<br>General: Support legacy use of<br>\bfdefault and \mddefault<br>(gh/306) .....                                                       | 667      |
| 2020-02-20 ltclass.dtx v1.3j<br>\endfilecontents: Fix missing quotes<br>around file name (gh/284) .....                                              | 1012 | 2020-03-19 ltfssdcl.dtx v3.0u<br>\document@select@group: fix for<br>(gnats/3357) .....                                                                              | 608      |
| 2020-02-24 ltfssbas.dtx v3.2f<br>\DeclareFontShape@: Drop surplus<br>“m” in series when defining<br>fontshape (gh/289) .....                         | 524  | 2020-03-19 ltfssini.dtx v3.1k<br>\DeclareFontSeriesDefault: Support<br>legacy use of \bfdefault and                                                                 |          |

|                                                                                                           |      |                                                                                                                |     |
|-----------------------------------------------------------------------------------------------------------|------|----------------------------------------------------------------------------------------------------------------|-----|
| \mddefault (gh/306) . . . . .                                                                             | 636  | \@vspace: Support calc syntax<br>(gh/152) . . . . .                                                            | 435 |
| \maybe@update@bfseries@defaults:<br>Support legacy use of \bfdefault<br>and \mddefault (gh/306) . . . . . | 643  | \@addvspace: Support calc syntax<br>(gh/152) . . . . .                                                         | 432 |
| \mdseries: Support legacy use of<br>\bfdefault and \mddefault<br>(gh/306) . . . . .                       | 644  | 2020-04-21 ltab.dtx v1.1r                                                                                      |     |
| 2020-04-06 ltfssini.dtx v3.1m                                                                             |      | \@itabcr: Support calc syntax<br>(gh/152) . . . . .                                                            | 843 |
| \bf@def@ult: Hook added (gh/306)                                                                          | 646  | \@yargarraycr: Support calc syntax<br>(gh/152) . . . . .                                                       | 851 |
| \maybe@update@bfseries@defaults:<br>Hook added (gh/306) . . . . .                                         | 643  | 2020-04-22 ltmiscen.dtx v1.1u                                                                                  |     |
| \maybe@update@mdseries@defaults:<br>Hook added (gh/306) . . . . .                                         | 644  | \@sverb: Drop spaces before \verb<br>delimiter (gh/327) . . . . .                                              | 781 |
| 2020-04-07 ltclass.dtx v1.3k                                                                              |      | 2020-04-22 ltoutenc.dtx v2.0p                                                                                  |     |
| \IfFormatAtLeastTF: Macro added;<br>also in rollback (gh/168) . . . . .                                   | 983  | General: y unicode value in tuenc.def                                                                          | 466 |
| \load@onefile@withoptions: Use<br>different method to ignore<br>unprocessed options (gh/22) . . . . .     | 1005 | 2020-04-29 lttextcomp.dtx v1.0d                                                                                |     |
| \ProcessOptions*: Use different<br>method to ignore unprocessed<br>options (gh/22) . . . . .              | 994  | General: Make all capital accents text<br>commands for hyperref (gh/332) . . . . .                             | 701 |
| \RequirePackageWithOptions: Use<br>different method to ignore<br>unprocessed options (gh/22) . . . . .    | 997  | 2020-05-02 ltfiles.dtx v1.2g                                                                                   |     |
| 2020-04-09 lftfloat.dtx v1.2d                                                                             |      | \@include: Support spaces in<br>filenames by enclosing the names<br>of .aux-files in quotes (gh/217) . . . . . | 452 |
| \@textsubscript: Set non-zero<br>baseline (gh/249) . . . . .                                              | 930  | \includeonly: Get rid of leading and<br>trailing spaces from the filename<br>(gh/217) . . . . .                | 450 |
| \textsubscript: Set non-zero baseline<br>(gh/249) . . . . .                                               | 929  | Improved support for spaces in<br>filenames (gh/217) . . . . .                                                 | 450 |
| 2020-04-13 ltfssdcl.dtx v3.0v                                                                             |      | Pass the filename to \@include by<br>value instead of by reference<br>(gh/217) . . . . .                       | 450 |
| \process@table: Small update for<br>speed. . . . .                                                        | 613  | 2020-05-05 ltxref.dtx v1.1n                                                                                    |     |
| 2020-04-13 ltfssini.dtx v3.1n                                                                             |      | \refstepcounter: record the counter<br>name in \currentcounter . . . . .                                       | 742 |
| \init@series@setup: Handling<br>\seriesdefault changes (gh/315)                                           | 642  | 2020-05-06 ltspace.dtx v1.3n                                                                                   |     |
| \seriesdefault@kernel: Handling<br>\seriesdefault changes (gh/315)                                        | 660  | General: Made softhyphen active in<br>TU engines . . . . .                                                     | 440 |
| 2020-04-21 ltmath.dtx v1.2f                                                                               |      | 2020-05-09 ltdefns.dtx v1.5j                                                                                   |     |
| \@yeqncr: Support calc syntax<br>(gh/152) . . . . .                                                       | 796  | \@if@DeclareRobustCommand: Added<br>\DeclareCommandCopy (gh/239) . . . . .                                     | 101 |
| 2020-04-21 ltmiscen.dtx v1.1t                                                                             |      | \@DeclareCommandCopy: Added<br>\DeclareCommandCopy (gh/239) . . . . .                                          | 98  |
| \@icentercr: Support calc syntax<br>(gh/152) . . . . .                                                    | 775  | 2020-05-11 ltdefns.dtx v1.5j                                                                                   |     |
| 2020-04-21 ltpictur.dtx v1.1o                                                                             |      | \@dischyp: Do not overwrite \-<br>under LuaTeX . . . . .                                                       | 110 |
| \@istackcr: Support calc syntax<br>(gh/152) . . . . .                                                     | 867  | 2020-05-15 ltdefns.dtx v1.5g                                                                                   |     |
| 2020-04-21 ltspace.dtx v1.3m                                                                              |      | \@typeout: Allow \par in the argument<br>(gh/335) . . . . .                                                    | 80  |
| \@hspace: Support calc syntax<br>(gh/152) . . . . .                                                       | 439  | 2020-05-19 ltfssaxes.dtx v1.0e                                                                                 |     |
| \@newline: Support calc syntax<br>(gh/152) . . . . .                                                      | 426  | \series@maybe@drop@one@cm: Need to<br>use \edef (gh/336) . . . . .                                             | 564 |
| \@vspace@calcify: Support calc<br>syntax (gh/152) . . . . .                                               | 426  | 2020-05-19 ltfssini.dtx v3.2a                                                                                  |     |
|                                                                                                           |      | \IfFontSeriesContextTF: Macros<br>added (gh/335) . . . . .                                                     | 651 |

|                                                                        |          |                                       |      |
|------------------------------------------------------------------------|----------|---------------------------------------|------|
| 2020-05-31 ltmiscen.dtx v1.1u                                          |          | 2020-07-27 ltsect.dtx v1.1e           |      |
| \centering: Added                                                      |          | \author: Don't make the command       |      |
| \ffinalhyphendemerits setting                                          |          | \long (gh/354) . . . . .              | 898  |
| (gh/247) . . . . .                                                     | 775      | \date: Don't make the command         |      |
| \raggedleft: Added                                                     |          | \long (gh/354) . . . . .              | 898  |
| \ffinalhyphendemerits setting                                          |          |                                       |      |
| (gh/247) . . . . .                                                     | 776      |                                       |      |
| \raggedright: Added                                                    |          |                                       |      |
| \ffinalhyphendemerits setting                                          |          |                                       |      |
| (gh/247) . . . . .                                                     | 776      |                                       |      |
| 2020-06-04 ltexpl.dtx v1.2c                                            |          | 2020-08-01 ltluatex.dtx v1.1p         |      |
| General: Define a local version of some                                |          | General: new_graf is exclusive . . .  | 63   |
| L <sup>A</sup> T <sub>E</sub> X 2 <sub>≤</sub> basic macros to support |          |                                       |      |
| package loading . . . . .                                              | 73       |                                       |      |
| 2020-06-04 ltfinal.dtx v2.2a                                           |          | 2020-08-02 ltluatex.dtx v1.1q         |      |
| General: Load ltexpl in ltdefns . . .                                  | 1191     | \newattribute: Move reset to 0 inside |      |
| 2020-06-05 ltclass.dtx v1.3l                                           |          | conditional . . . . .                 | 49   |
| \currnamestack: Added                                                  |          | \newluabytecode: Move reset to 0      |      |
| \@expl@pop@filename@0@ . . . . .                                       | 981      | inside conditional . . . . .          | 52   |
| Added \@expl@push@filename@0@                                          |          | \newluachunkname: Move reset to 0     |      |
| and                                                                    |          | inside conditional . . . . .          | 52   |
| \@expl@push@filename@aux@0@ . . . . .                                  | 980      | \newluafunction: Move reset to 0      |      |
| 2020-06-05 ltfiles.dtx v1.2h                                           |          | inside conditional . . . . .          | 51   |
| \document: Added hook to load                                          |          | \newwhatsit: Move reset to 0 inside   |      |
| l3backend code . . . . .                                               | 444      | conditional . . . . .                 | 51   |
| 2020-06-10 ltluatex.dtx v1.1n                                          |          | 2020-08-08 ltclass.dtx v1.3m          |      |
| General: Define                                                        |          | \endfilecontents: define              |      |
| \@gobble/\@firstofone even for                                         |          | \q@curr@file directly as the          |      |
| L <sup>A</sup> T <sub>E</sub> X to allow early loading. . . . .        | 47       | quotes have already been removed      |      |
| 2020-07-04 ltoutenc.dtx v2.0q                                          |          | (gh/220) . . . . .                    | 1011 |
| General: Implement \remove@tlig in                                     |          | 2020-08-10 ltluatex.dtx v1.1r         |      |
| LuaTeXwithout font reloading . . . . .                                 | 498      | General: Load ltluatex Lua module     |      |
| 2020-07-08 ltexpl.dtx v1.2d                                            |          | during format building . . . . .      | 52   |
| General: Add a last-minute hook for                                    |          | 2020-08-15 ltpictur.dtx v1.2a         |      |
| expl3 . . . . .                                                        | 74       | \@defaultunitset: Macro added . . .   | 864  |
| 2020-07-08 ltfinal.dtx v2.2b                                           |          | 2020-08-19 ltdefns.dtx v1.5k          |      |
| General: Add a last-minute hook for                                    |          | \carcube: Made \long for              |      |
| expl3 . . . . .                                                        | 1180     | \NewCommandCopy . . . . .             | 83   |
| 2020-07-27 ltmath.dtx v1.2g                                            |          | \robust@command@act: Made             |      |
| \cases: Don't make the command                                         |          | \robust@command@act (was              |      |
| \long (gh/354) . . . . .                                               | 789      | \declare@command@copy) more           |      |
| \matrix: Don't make the command                                        |          | generic . . . . .                     | 96   |
| \long (gh/354) . . . . .                                               | 789      | \ShowCommand: Added \ShowCommand      |      |
| \pmatrix: Don't make the command                                       |          | (gh/373) . . . . .                    | 100  |
| \long (gh/354) . . . . .                                               | 789      | 2020-08-19 ltexpl.dtx v1.2e           |      |
| 2020-07-27 ltoutenc.dtx v2.0r                                          |          | General: Add                          |      |
| \use@text@encoding: Don't make the                                     |          | \@expl@cs@{thing}@spec@0@N for        |      |
| command \long (gh/354) . . . . .                                       | 475, 476 | \ShowCommand (gh/373) . . . . .       | 77   |
| 2020-07-27 ltpage.dtx v1.0m                                            |          | Add \@expl@cs@to@str@0@N and          |      |
| \markright: Don't make the command                                     |          | \@expl@str@if@eq@0@nnTF for           |      |
| \long (gh/354) . . . . .                                               | 970, 971 | \NewCommandCopy (gh/239) . . . . .    | 76   |
| 2020-07-27 ltpictur.dtx v1.1p                                          |          | 2020-08-20 lplain.dtx v2.3d           |      |
| \linethickness: Don't make the                                         |          | \alloc@: Define \alloc@ in terms of   |      |
| command \long (gh/354) . . . . .                                       | 867      | \@alloc . . . . .                     | 21   |

|                                                                                                                                                   |                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2020-08-21 ltdefns.dtx v1.5m<br>\MakeRobust: Make \MakeRobust produce the same command structure as \DeclareRobustCommand . . . . . 92            | 2020-09-09 ltshipout.dtx v1.0b<br>\__shipout_picture_overlay:n: Prevent overfull box warnings (gh/387) . . . . . 1079                                                             |
| 2020-08-21 ltxexpl.dtx v1.2d<br>General: Dropped unused command . . . . . 73                                                                      | 2020-09-26 ltfinal.dtx v2.2j<br>General: Load first aid file if existing 1192                                                                                                     |
| 2020-08-21 ltffiles.dtx v1.2i<br>General: Integration of new hook management interface . . . . . 442                                              | 2020-09-30 ltfssini.dtx v3.2d<br>\maybe@update@bfseries@defaults:<br>\bfdefault@previous not<br>\bfseries@previous (gh/395) . . . . . 643                                         |
| 2020-08-21 ltffinal.dtx v2.2i<br>General: Integration of new hook management interface . . . . . 1174                                             | \mdseries: \mddefault@previous not<br>\mdseries@previous (gh/395) . . . . . 644                                                                                                   |
| 2020-08-21 ltfssaxes.dtx v1.0g<br>General: Integration of new hook management interface . . . . . 571                                             | 2020-10-01 ltclass.dtx v1.3r<br>\@pr@videpackage: Allow for package substitution . . . . . 988                                                                                    |
| 2020-08-21 ltfssini.dtx v3.2b<br>\bf@def@ult: Integration of new hook management interface . . . . . 646                                          | 2020-10-01 ltsect.dtx v1.1e<br>\addcontentsline: add a fourth argument for better hyperref compatibility . . . . . 907                                                            |
| \mdseries@defaults: Integration of new hook management interface . . . . . 649                                                                    | 2020-10-04 ltffiles.dtx v1.2j<br>\@include: Quotes around the aux file name removed, they are not needed and upset BibTeX (gh/400) . . . . . 452                                  |
| \maybe@update@bfseries@defaults:<br>Integration of new hook management interface . . . . . 643                                                    | 2020-10-04 lthooks.dtx v1.0d<br>General: Definition \AddToHookNext was supposed to be for \AddToHook vice versa (gh/401) . . . . . 301                                            |
| \maybe@update@mdseries@defaults:<br>Integration of new hook management interface . . . . . 644                                                    | 2020-10-08 ltclass.dtx v1.3s<br>\@currnamestack: Added missing 2020/02/02 \IncludeInRelease . . . . . 980                                                                         |
| \reset@font: Integration of new hook management interface . . . . . 657                                                                           | 2020-10-11 ltclass.dtx v1.3t<br>\load@onefilewithoptions: Restore \currpkg@reqd after finished loading a package file (gh/408). . . . . 1003                                      |
| \rmfamily: Integration of new hook management interface . . . . . 648                                                                             | 2020-10-18 ltclass.dtx v1.3t<br>\PassOptionsToClass: Drop path from \input@path (gh/414). . . . . 990                                                                             |
| \ttfamily: Integration of new hook management interface . . . . . 648                                                                             | 2020-10-23 ltmiscen.dtx v1.1w<br>\enddocument: Make enddocument/afteraux one-time 762                                                                                             |
| 2020-08-21 ltmiscen.dtx v1.1v<br>General: Integration of new hook management interface . . . . . 760                                              | 2020-10-26 ltmiscen.dtx v1.1x<br>\kernel@before@enddocument:<br>\enddocument should always start out in vmode (gh/385) . . . . . 765                                              |
| 2020-08-21 ltoutput.dtx v1.4f<br>\begin{indvibox}: Integration of new hook management interface . . . . . 1099                                    | 2020-11-09 ltclass.dtx v1.3u<br>\pkgcls@rollbackdate@error:<br>Change help text because the package may have existed then — there is just no rollback data (gh/423). . . . . 1024 |
| 2020-08-23 ltxref.dtx v1.1o<br>\refstepcounter: add default definition of \currentcounter . . . . . 742                                           | 2020-11-09 ltmath.dtx v1.2h<br>\expl@str@map@function@NN and \expl@neg@ignorespace@negmedspace and \negthickspace have been only in                                               |
| 2020-09-06 ltclass.dtx v1.3q<br>\load@onefilewithoptions: Save \currpkg@reqd so that we don't lose track of package substitutions. . . . . 1002   |                                                                                                                                                                                   |
| 2020-09-06 ltdefns.dtx v1.5n<br>\char@if@alph: Macro added . . . . . 109                                                                          |                                                                                                                                                                                   |
| 2020-09-06 ltxexpl.dtx v1.2f<br>General: Add \expl@str@map@function@NN and \expl@neg@ignorespace@negmedspace and \negthickspace have been only in |                                                                                                                                                                                   |
| \@expl@string@makeletter (gh/386) . . . . . 77                                                                                                    |                                                                                                                                                                                   |

|                                                                                                                                                                                                                     |              |                                                                                                                                                  |                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| amsmath, so we need to undefine<br>for rollback (gh/423) . . . . .                                                                                                                                                  | <b>792</b>   | 2020-12-04 ltfilehook.dtx v1.0d<br>\undeclare@file@substitution:<br>Don't drop file substitution<br>commands on rollback . . . . .               | <b>1048</b>     |
| 2020-11-20 ltclass.dtx v1.3u<br>\@currpath: Macro added . . . . .                                                                                                                                                   | <b>979</b>   | 2020-12-04 ltfssaxes.dtx v1.0h<br>General: Reorganized the rollback<br>data . . . . .                                                            | <b>551</b>      |
| \@kernel@\currpathstack: Macro<br>added . . . . .                                                                                                                                                                   | <b>982</b>   | \fontseries: Distangle series and<br>shape update (gh/444) . . . . .                                                                             | <b>560</b>      |
| \load@onefile@withoptions: Copy<br>option list to the requested<br>package. . . . .                                                                                                                                 | <b>1004</b>  | \fontshape: Distangle series and<br>shape update (gh/444) . . . . .                                                                              | <b>568</b>      |
| \PassOptionsToClass: Copy option<br>list to the requested package. . . . .                                                                                                                                          | <b>990</b>   | \fontshapeforce: Distangle series and<br>shape update (gh/444) . . . . .                                                                         | <b>568</b>      |
| \ProvidesPackage: Use string<br>comparison instead of \ifx . . . . .                                                                                                                                                | <b>987</b>   | 2020-12-04 ltfssini.dtx v3.2f<br>General: Adjust start values for series<br>and shape (gh/444) . . . . .                                         | <b>659</b>      |
| 2020-11-20 ltcmd.dtx v1.0a<br>General: Initial version derived from<br>xparses.dtx . . . . .                                                                                                                        | <b>113</b>   | 2020-12-10 ltbibl.dtx v1.1s<br>\nocite: Delay any \nocite in the<br>preamble instead of raising an<br>error . . . . .                            | <b>940</b>      |
| 2020-11-20 ltfilehook.dtx v1.0d<br>\unqu@tefilef@und: Move loading to<br>\@input@file@exists@with@hooks<br>and expand \@filef@und to avoid<br>getting the wrong file name in the<br>case of a substitution. . . . . | <b>1045</b>  | 2020-12-10 ltfssbas.dtx v3.2h<br>\usefont: Drop "m" if the series value<br>is a member of a fixed list and<br>issue warning if doing it (gh/453) | <b>532</b>      |
| 2020-11-23 ltshipout.dtx v1.0d<br>\_shipout\_execute\_cont:: Check<br>for both kernel and user hook<br>(gh/431) . . . . .                                                                                           | <b>1069</b>  | 2020-12-14 ltclass.dtx v1.3v<br>\currnamestack: Removed<br>\@expl@@@hook@curr@name@push@on<br>. . . . .                                          | <b>980</b>      |
| \_shipout\_execute\_main\_cont:Nnnn:<br>Check for both kernel and user<br>hook (gh/431) . . . . .                                                                                                                   | <b>1071</b>  | 2020-12-18 ltexpl.dtx v1.2h<br>\@kernel@after@enddocument@afterlastpage:<br>Define kernel \enddocument hooks<br>early . . . . .                  | <b>73</b>       |
| 2020-11-24 ltexpl.dtx v1.2g<br>General: Support for roll forward<br>(gh/434) . . . . .                                                                                                                              | <b>75-77</b> | 2020-12-22 ltfssaxes.dtx v1.0h<br>\delayed@merge@font@series:<br>Distangle series and shape update<br>(gh/444) . . . . .                         | <b>562, 563</b> |
| 2020-11-24 ltfilehook.dtx v1.0d<br>General: Support for roll forward<br>(gh/434) . . . . .                                                                                                                          | <b>1043</b>  | \delayed@merge@font@shape:<br>Distangle series and shape update<br>(gh/444) . . . . .                                                            | <b>569</b>      |
| 2020-11-24 lthooks.dtx v1.0f<br>\_hook_end_document_label_check::<br>Support for roll forward (gh/434)                                                                                                              | <b>237</b>   | 2020-12-22 lfsstrc.dtx v3.0n<br>\selectfont: Execute delayed series<br>and shape updates (gh/444) . . .                                          | <b>576</b>      |
| 2020-11-24 ltshipout.dtx v1.0d<br>General: Support for roll forward<br>(gh/434) . . . . .                                                                                                                           | <b>1083</b>  | 2021-01-07 ltfilehook.dtx v1.0e<br>General: Added rollback for this case<br>to avoid spurious errors (part of<br>gh/463) . . . . .               | <b>1057</b>     |
| \AtBeginDvi: Support for roll forward<br>(gh/434) . . . . .                                                                                                                                                         | <b>1082</b>  | \unqu@tefilef@und: Restore<br>\CurrentFile(Path)(Used) after<br>the input (gh/464) . . . . .                                                     | <b>1046</b>     |
| 2020-11-25 ltdefns.dtx v1.5o<br>\carcube: Added missing latexrelease<br>entry . . . . .                                                                                                                             | <b>83</b>    | 2021-01-07 lthooks.dtx v1.0h<br>\_hook_strip_double_slash:w:<br>Assume hook name has at least<br>three nonempty parts (gh/464) .                 | <b>249</b>      |
| 2020-12-02 ltluatex.dtx v1.1s<br>General: Fix return value of list<br>callbacks . . . . .                                                                                                                           | <b>65</b>    |                                                                                                                                                  |                 |
| 2020-12-03 lfsstrc.dtx v3.0m<br>\selectfont: Install a hook in<br>\selectfont (gh/444) . . . . .                                                                                                                    | <b>577</b>   |                                                                                                                                                  |                 |

|                                                                                                                                                                                             |      |                                                                                                              |               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------|---------------|
| \_\_hook\_tl\_set:cn: Manually define some l3tl commands to work around expl3 changes . . . . .                                                                                             | 224  | \end\verb visible space: Undo the internals for robust \begin and \end in rollback (gh/494) . . . . .        | 773           |
| 2021-01-08 ltshipout.dtx v1.0f                                                                                                                                                              |      | 2021-02-10 ltboxes.dtx v1.4b                                                                                 |               |
| \_\_shipout\_execute\_cont:: Added another kernel hook for more flexibility (cf. <a href="https://github.com/pgf-tikz/pgf/issu">https://github.com/pgf-tikz/pgf/issu</a> ue#1900) . . . . . | 1069 | \@mpfootnotetext: Explicitly run \par in support for paragraph tagging . . . . .                             | 832           |
| 2021-01-10 ltshipout.dtx v1.0g                                                                                                                                                              |      | \@footnotetext: Explicitly run \par at the end of footnote text in preparation for paragraph hooks . . . . . | 930           |
| \@kernel@after@shipout@background:                                                                                                                                                          |      | 2021-02-15 ltfsdcl.dtx v3.0w                                                                                 |               |
| Internal hook                                                                                                                                                                               |      | \document@select@group: fix for (gh/501) . . . . .                                                           | 608           |
| \@kernel@after@shipout@background added . . . . .                                                                                                                                           | 1073 | 2021-02-16 ltfloat.dtx v1.2f                                                                                 |               |
| \RawShipout: Macro added . . . . .                                                                                                                                                          | 1072 | \footref: \footref added . . . . .                                                                           | 933           |
| 2021-01-19 ltshipout.dtx v1.0h                                                                                                                                                              |      | 2021-02-17 ltoutenc.dtx v2.0t                                                                                |               |
| \_\_shipout\_run\_firstpage\_hook:: Handling of firstpage hook altered . . . . .                                                                                                            | 1073 | General: Adjust values for \textasteriskcentered To match TS1 definition (gh/502) . . . . .                  | 503           |
| 2021-01-21 ltclass.dtx v1.3w                                                                                                                                                                |      | Special definition for \textasteriskcentered when missing in TS1 (gh/502) . . . . .                          | 494           |
| \@kernel@currpathstack: Add empty entry for latexrelease . . . . .                                                                                                                          | 982  | 2021-02-18 ltclass.dtx v1.3x                                                                                 |               |
| 2021-01-21 ltexpl.dtx v1.3a                                                                                                                                                                 |      | \@fileswithoptions: save raw class option list (gh/85) . . . . .                                             | 999           |
| General: Move xparse rollback code to ltcmd.dtx . . . . .                                                                                                                                   | 76   | \@remove@eq@value: macro added (gh/85) . . . . .                                                             | 991           |
| 2021-01-21 ltfinal.dtx v2.2l                                                                                                                                                                |      | \@use@option: value from unused option list (gh/85) . . . . .                                                | 994           |
| General: Load glyptounicode.tex for pdfTeX . . . . .                                                                                                                                        | 1182 | \@optionNotUsed: value from unused option list (gh/85) . . . . .                                             | 991           |
| 2021-01-22 ltshipout.dtx v1.0i                                                                                                                                                              |      | \PassOptionsToClass: save raw option lists (gh/85) . . . . .                                                 | 990           |
| \_\_shipout\_finalize\_box:: Add pre\_ shipout\_ filter Lua callback . . . . .                                                                                                              | 1068 | 2021-02-19 ltoutenc.dtx v2.0u                                                                                |               |
| 2021-01-24 ltexpl.dtx v1.3a                                                                                                                                                                 |      | General: Add \textnonbreakinghyphen, \textfiguredash and \texthorizontalbar (gh/404) . . . . .               | 483, 487, 502 |
| General: Define expl3 hooks conditionally . . . . .                                                                                                                                         | 73   | 2021-02-25 ltfinal.dtx v2.2m                                                                                 |               |
| 2021-01-31 ltfilehook.dtx v1.0f                                                                                                                                                             |      | General: Improve speed of ToUnicode everyjob loading code . . . . .                                          | 1182          |
| \@curr@file@reqd: set \protect to \string gh/481 . . . . .                                                                                                                                  | 1049 | 2021-03-03 ltclass.dtx v1.3y                                                                                 |               |
| 2021-02-03 ltfloat.dtx v1.2e                                                                                                                                                                |      | \endfilecontents: Fix overwrite check for files with UTF-8 (gh/415) . . . . .                                | 1011          |
| \@savemarbox: Explicitly end with \par (gh/489) . . . . .                                                                                                                                   | 924  | 2021-03-05 ltclass.dtx v1.3z                                                                                 |               |
| 2021-02-04 ltboxes.dtx v1.4b                                                                                                                                                                |      | \ProcessOptions*: modify so braces to not give errors (gh/513) . . . . .                                     | 992           |
| \color@endbox: Always add the color groups (gh/488) . . . . .                                                                                                                               | 822  | 2021-03-12 ltfiles.dtx v1.2k                                                                                 |               |
| 2021-02-08 ltfilehook.dtx v1.0g                                                                                                                                                             |      | \IfFileExists@: Allow unbalanced conditionals (gh/530) . . . . .                                             | 457           |
| \unqu@tefilef@und: Undo the internal for robust \InputIfFileExists in rollback (gh/494) . . . . .                                                                                           | 1047 |                                                                                                              |               |
| 2021-02-08 ltmiscen.dtx v1.1y                                                                                                                                                               |      |                                                                                                              |               |
| \begin: Undo the internals for robust \begin and \end in rollback (gh/494) . . . . .                                                                                                        | 770  |                                                                                                              |               |

|                                 |                                             |      |                                   |                                           |      |
|---------------------------------|---------------------------------------------|------|-----------------------------------|-------------------------------------------|------|
| 2021-03-18 ltcmd.dtx v1.0b      | General: Use \NewModuleRelease. . .         | 113  | 2021-04-19 ltcmd.dtx v1.0d        | \_cmd\_cmd\_type\_cases:NnnnnnF:          |      |
| 2021-03-18 ltfilehook.dtx v1.0h | \_filehook\_file\_pop\_assign:nnnn:         |      | Renamed                           | \_cmd\_cmd\_if\_xparse:NTF to             |      |
|                                 | Define \g_@@_input_file_seq to              |      | \_kernel\_cmd\_if\_xparse:NTF for | cross-module usage . . . . .              | 181  |
|                                 | avoid losing data when rolling              |      |                                   |                                           |      |
|                                 | back. . . . .                               | 1043 |                                   |                                           |      |
| 2021-03-18 ltfssaxes.dtx v1.0i  | General: Fix rollforward definition. . .    | 561  | 2021-04-20 ltexpl.dtx v1.3c       | \@kernel@after@enddocument@afterlastpage: |      |
| 2021-03-18 ltfssini.dtx v3.2g   | General: Add legacy hook definitions        |      | Don't empty kernel hooks on       | rollback . . . . .                        | 73   |
|                                 | for rollback. . . . .                       | 649  |                                   |                                           |      |
| 2021-03-18 lthooks.dtx v1.0i    | \_hook\_end\_document\_label\_check::       |      | 2021-04-20 ltfilehook.dtx v1.0i   | \@curr@file@reqd: Make expand to          |      |
|                                 | Only add top-level if not already           |      | a string (tracks change in        | l3kernel) . . . . .                       | 1049 |
|                                 | there. . . . .                              | 237  |                                   |                                           |      |
|                                 | Remove the (empty) "top-level"              |      | 2021-04-26 ltfssbas.dtx v3.2i     | \usefont: Unconditionally switch to       |      |
|                                 | from \@currnamestack. . . . .               | 237  | the requested font face (gh/444)  | 532                                       |      |
|                                 | General: Use \NewModuleRelease. . .         | 222  | 2021-04-26 ltfssini.dtx v3.2h     | \reset@font: Unconditionally switch       |      |
| 2021-03-18 ltvers.dtx v1.1f     | \@check@IncludeInRelease: Add               |      | to the requested font face        | (gh/444) . . . . .                        | 657  |
|                                 | support for usage in                        |      | 2021-04-26 lfsstrc.dtx v3.0o      | \selectfont: Unset the forced series      |      |
|                                 | \NewModuleRelease . . . . .                 | 39   | boolean when reaching             | \selectfont (gh/444) . . . . .            | 577  |
|                                 | \new@moduledate: Added                      |      | 2021-04-29 lthooks.dtx v1.0m      | \ActivateGenericHook: Add                 |      |
|                                 | \NewModuleRelease. . . . .                  | 40   | \ProvideHook etc. . . . .         | 296                                       |      |
| 2021-03-19 lttextcomp.dtx v1.0e | General: Use \NewModuleRelease . . .        | 695  | 2021-04-29 ltoutenc.dtx v2.0v     | General: Add composites for               |      |
| 2021-03-26 ltplain.dtx v2.3e    | \@unused: Allocate \@inputcheck and         |      | \ae/\AE/\æ/\鏗 (gh/552) . . . . .  | 507                                       |      |
|                                 | \@unused early so that they are             |      | 2021-05-18 ltclass.dtx v1.4b      | \@raw@classoptionslist: Initialise to     |      |
|                                 | before expl3 allocates more                 |      | \relax to match                   | \@classoptionslist . . . . .              | 979  |
|                                 | streams (gh/538) . . . . .                  | 23   | 2021-05-24 ltcmd.dtx v1.0e        | General: Use \msg_... instead of          |      |
| 2021-03-27 ltclass.dtx v1.4a    | \@currnamestack: Do not completely          |      | \_kernel\_msg... . . . . .        | 113                                       |      |
|                                 | roll back if expl3 is loaded. . . .         | 981  | 2021-05-24 ltcmdhooks.dtx v1.0b   | General: Use \msg_... instead of          |      |
| 2021-04-16 ltvers.dtx v1.1g     | \new@moduledate: \NewModuleRelease          |      | \_kernel\_msg... . . . . .        | 308                                       |      |
|                                 | with the same arguments as                  |      | 2021-05-24 ltfilehook.dtx v1.0k   | General: Use \msg_... instead of          |      |
|                                 | \IncludeInRelease. . . . .                  | 40   | \_kernel\_msg... . . . . .        | 1041                                      |      |
| 2021-04-17 ltfiles.dtx v1.2m    | \@kernel@after@begindocument:               |      | 2021-05-24 lthooks.dtx v1.0n      | General: Use \msg_... instead of          |      |
|                                 | Move                                        |      | \_kernel\_msg... . . . . .        | 222                                       |      |
|                                 | \@kernel@before@begindocument               |      | 2021-05-24 ltpara.dtx v1.0g       | General: Use \msg_... instead of          |      |
|                                 | and                                         |      | \_kernel\_msg... . . . . .        | 409                                       |      |
|                                 | \@kernel@after@begindocument                |      | 2021-05-26 ltdefns.dtx v1.5p      | \MakeRobust: Normalize error message      |      |
|                                 | init earlier so that other modules          |      |                                   | in \MakeRobust . . . . .                  | 92   |
|                                 | can write to the hooks . . . . .            | 447  |                                   |                                           |      |
| 2021-04-18 ltluatex.dtx v1.1t   | General: input_level_string added .         | 63   |                                   |                                           |      |
|                                 |                                             |      |                                   |                                           |      |
| 2021-04-18 ltplain.dtx v2.3f    | \loggingall: 3 . . . . .                    | 33   |                                   |                                           |      |
|                                 | Drop pre- $\epsilon$ -TeX support . . . . . | 33   |                                   |                                           |      |
|                                 | \tracingnone: 3 . . . . .                   | 34   |                                   |                                           |      |
|                                 | Drop pre- $\epsilon$ -TeX support . . . . . | 34   |                                   |                                           |      |

|                               |                                                                                                |      |                                                                                                                       |      |
|-------------------------------|------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------|------|
| 2021-06-03 ltclass.dtx v1.4c  | \@kernel@currpathstack: Take care of \@kernel@currpathstack when rolling back/forward. . . . . | 982  | \@unusedoptionlist: Drop \@onlypreamble . . . . .                                                                     | 979  |
| 2021-06-04 ltcmd.dtx v1.0f    | General: Normalize various error messages . . . . .                                            | 183  | \IfFormatAtLeastTF: Drop \@onlypreamble . . . . .                                                                     | 983  |
| 2021-06-05 ltmiscen.dtx v1.1z | \@@sverb: Normalize error message .                                                            | 781  | 2021-07-20 ltcmdhooks.dtx v1.0c                                                                                       |      |
| 2021-06-06 ltclass.dtx v1.4c  | \@loadwithoptions: handle raw options for gh/580 . . . . .                                     | 996  | \_\_hook\_patch_DeclareRobustCommand:Nnn: Use \robust@command@chk@safe before \@if@newcommand. . . . .                | 312  |
|                               | \load@onefile@withoptions: Copy raw options for gh/580 . . . . .                               | 1004 | \_\_hook_gremove_code:nn: Do not queue removals (gh/625) . . . . .                                                    | 251  |
|                               | \PassOptionsToClass: apply \expandafter to raw options for gh/580 . . . . .                    | 990  | \_\_hook_prop_gput_labeled_do:Nnnn: Do not queue removals (gh/625)                                                    | 239  |
| 2021-06-09 ltclass.dtx v1.4b  | \endfilecontents: Use \@latex@note@no@line to display the information . . . . .                | 1011 | 2021-07-23 ltclass.dtx v1.4e                                                                                          |      |
| 2021-06-09 lterror.dtx v1.2r  | \@latex@note@no@line: Macros added . . . . .                                                   | 393  | \load@onefile@withoptions: Make class/name/after a one-time hook . . . . .                                            | 1005 |
| 2021-06-09 ltfsbas.dtx v3.2j  | \DeclareFontShape@: Improve information message . . . . .                                      | 524  | Make class/name/before a one-time hook . . . . .                                                                      | 1004 |
| 2021-07-08 ltcnts.dtx v1.1m   | \counterwithin: New implementation for \counterwithout and \counterwithin . . . . .            | 516  | Make package/name/after a one-time hook . . . . .                                                                     | 1005 |
| 2021-07-11 lterror.dtx v1.2s  | \PackageNoteNoLine: Provide \ClassNote and \PackageNote .                                      | 392  | Make package/name/before a one-time hook . . . . .                                                                    | 1004 |
| 2021-07-12 ltclass.dtx v1.4d  | \@fileswithoptions: add \unexpanded . . . . .                                                  | 999  | 2021-07-23 ltfiles.dtx v1.2n                                                                                          |      |
|                               | \ClearHookNext: Macro added . . . . .                                                          | 297  | \include: Make include/name/after a one-time hook . . . . .                                                           | 452  |
| 2021-07-16 ltplain.dtx v2.3g  | General: Use 2 as default value for \tracinglostchars . . . . .                                | 24   | Make include/name/before a one-time hook . . . . .                                                                    | 452  |
| 2021-07-19 ltclass.dtx v1.4e  | \@classoptionslist: Drop \@onlypreamble . . . . .                                              | 979  | Make include/name/end a one-time hook . . . . .                                                                       | 452  |
|                               | \@ifclasslater: Drop \@onlypreamble . . . . .                                                  | 983  | 2021-07-27 lthooks.dtx v1.0o                                                                                          |      |
|                               | \@ifclassloaded: Drop \@onlypreamble . . . . .                                                 | 983  | \ClearHookNext: Macro added . . . . .                                                                                 | 297  |
|                               | \@ifclasswith: Drop \@onlypreamble . . . . .                                                   | 985  | \hook_gclear_next_code:n: Macro made public . . . . .                                                                 | 283  |
|                               | \@ifl@ter: Drop \@onlypreamble .                                                               | 984  | 2021-07-28 ltsect.dtx v1.1f                                                                                           |      |
|                               | \@pkgextension: Drop \@onlypreamble . . . . .                                                  | 979  | \contentsline: Pick up four arguments (gh/633) . . . . .                                                              | 908  |
|                               | \@optionlist: Drop \@onlypreamble .                                                            | 983  | 2021-07-30 ltcmd.dtx v1.0d                                                                                            |      |
|                               |                                                                                                |      | \_\_cmd_cmd_type_cases:NnnnnF: Added \@@_cmd_type_cases:NnnnnF for \NewCommandCopy and \ShowCommand support . . . . . | 181  |
|                               |                                                                                                |      | 2021-07-31 ltoutput.dtx v1.4e                                                                                         |      |
|                               |                                                                                                |      | \fl@tracemessage: Enable display when doing \tracefloatvals .                                                         | 1146 |
|                               |                                                                                                |      | 2021-07-31 ltoutput.dtx v1.4g                                                                                         |      |
|                               |                                                                                                |      | \ShowFloat: Macro added . . . . .                                                                                     | 1144 |
|                               |                                                                                                |      | 2021-08-02 lthooks.dtx v1.0o                                                                                          |      |
|                               |                                                                                                |      | \ActivateGenericHook: Change name . . . . .                                                                           | 296  |
|                               |                                                                                                |      | \DisableGenericHook: Change name                                                                                      | 296  |

|                                                                                                                                                                                   |                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 2021-08-07 ltcmd.dtx v1.0g<br>\_\_cmd\_add\_grabber:N: Replicate argument processors for all embellishments (gh/639) . . . . . 140                                                | 2021-08-27 ltfilehook.dtx v1.0l<br>\_\_filehook\_file\_pop\_assign:nnnn: Internal message name changes 1043                              |
| \_\_cmd\_add\_type_E:w: Replicate argument processors for all embellishments (gh/639) . . . . . 138                                                                               | \_\_filehook\_file\_subst\_cycle\_error:cN: Use \msg_... not \_\_kernel\_msg_... . . . . . 1054                                          |
| 2021-08-08 ltfinal.dtx v2.2p<br>\IfPDFManagementActiveTF: Default definition added (gh/640) . . . . . 1192                                                                        | 2021-08-27 lthooks.dtx v1.0q<br>General: Internal message name changes . . . . . 292                                                     |
| 2021-08-10 ltvers.dtx v1.1h<br>\@check@IncludeInRelease: Add error to aid debugging . . . . . 39                                                                                  | 2021-08-27 ltpara.dtx v1.0i<br>General: Internal message name changes . . . . . 417                                                      |
| 2021-08-11 lltuatax.dtx v1.1u<br>General: Define missing local function 54                                                                                                        | 2021-08-30 ltcmd.dtx v1.0h<br>General: Added support for \NewCommandCopy . . . . . 144                                                   |
| 2021-08-20 lterror.dtx v1.2t<br>\@badend: Improve \@badend error message (gh/587) . . . . . 396                                                                                   | Added support for \ShowCommand 150                                                                                                       |
| 2021-08-20 lthooks.dtx v1.0p<br>General: Added deprecation warnings for old generic hook commands (gh/638) . . . . . 299                                                          | 2021-09-03 ltoutput.dtx v1.4h<br>\ShowFloat: Renamed, original name never distributed . . . . . 1144                                     |
| Documentation updates for generic hook commands (gh/638) . . . . . 195                                                                                                            | 2021-09-06 ltfinal.dtx v2.2q<br>\@cuclclist: Correctly upper and lowercase \ij and \IJ (gh/658) 1190                                     |
| Renames of generic hook commands (gh/638) . . . . . 233                                                                                                                           | 2021-09-06 lthooks.dtx v1.0r<br>\_\_hook_if_execute_immediately:n: Macro added (gh/606) . . . . . 288                                    |
| Section on generic hooks added (gh/638) . . . . . 212                                                                                                                             | \_\_hook_prop_gput_labeled_do:Nnn: Use dedicated conditional (gh/606) . . . . . 239                                                      |
| 2021-08-25 ltclass.dtx v1.4f<br>General: Standardise generic hook names (gh/648) . . . . . 974                                                                                    | \_\_hook_use_once_clear:n: Clean up after \UseOneTimeHook (gh/606) 288                                                                   |
| \load@onefile@withoptions: Declare non-generic package and class hooks . . . . . 1005                                                                                             | \hook_use_once:nnw: Clean up after \UseOneTimeHook (gh/606) . . . . . 286                                                                |
| 2021-08-25 ltcmdhooks.dtx v1.0d<br>\_\_hook_try_put_cmd_hook:w:<br>Simplify generic hook detection . . . . . 309                                                                  | 2021-09-10 ltfssini.dtx v3.2i<br>\bfseries: Do delayed changes to \bfdefault in a separate macro for better reuse (gh/663) . . . . . 643 |
| 2021-08-25 ltfilehook.dtx v1.0l<br>\unqu@tefilef@nd: Declare non-generic file hooks . . . . . 1046                                                                                | \DeclareFontSeriesDefault: Do delayed changes to \bfdefault or \mddefault first (gh/663) . . . . . 636                                   |
| 2021-08-25 ltfiles.dtx v1.2o<br>\@include: Declare non-generic include hooks . . . . . 453                                                                                        | \maybe@update@bfseries@defaults: Do delayed changes to \bfdefault in a separate macro for better reuse (gh/663) . . . . . 643            |
| Standardise generic hook names (gh/648) . . . . . 452                                                                                                                             | \maybe@update@mdseries@defaults: Do delayed changes to \mddefault in a separate macro for better reuse (gh/663) . . . . . 644            |
| 2021-08-25 lthooks.dtx v1.0p<br>\_\_hook_try_declarng_generic_next_hook:nn\mdseries: Do delayed changes to \mddefault in a separate macro for better reuse (gh/663) . . . . . 644 | 2021-09-12 lfntcmd.dtx v3.5a<br>\check@nocorr@: use \unexpanded to make # safe . . . . . 690                                             |
| 2021-08-27 ltcmd.dtx v1.0h<br>\NewDocumentEnvironment: Check for end-of-environment command . . . . . 189                                                                         |                                                                                                                                          |

|                                 |                                                                                                       |     |                                 |                                                                                                                                 |          |
|---------------------------------|-------------------------------------------------------------------------------------------------------|-----|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------|----------|
| 2021-09-12 ltoutenc.dtx v2.0w   | General: Move zero skip between i and j for hyphenation (gh/658) . . . . .                            | 484 | 2021-10-19 ltpara.dtx v1.0k     | General: Remove content from \tex_everypar:D on rollback . . . . .                                                              | 417      |
| 2021-09-18 ltpara.dtx v1.0i     | \para_end:: Use skip rather than kern as guard. . . . .                                               | 415 | 2021-10-20 ltcmdhooks.dtx v1.0f | \__hook_make_prefixes:w: Correct patching by expansion+redefinition when the macro contains a parameter token (gh/697). . . . . | 313      |
| 2021-09-26 ltfsdcl.dtx v3.0x    | \c@localmathalphabets: Counter added for (gh/676) . . . . .                                           | 607 | 2021-11-17 ltluatex.dtx v1.1w   | General: hpack_ quality is exclusive . . . . .                                                                                  | 63       |
|                                 | \document@select@group: Test if we should freeze the version (gh/676)                                 | 607 |                                 | Never pass on true return values for list callbacks . . . . .                                                                   | 65       |
|                                 | \freeze@math@version: Macro added for (gh/676) . . . . .                                              | 610 |                                 | vpack_ quality is exclusive . . . . .                                                                                           | 63       |
| 2021-09-28 ltcmdhooks.dtx v1.0e | \__hook_make_prefixes:w: Make patching of commands a global operation (gh/674) . . . . .              | 318 | 2021-11-30 ltclass.dtx v1.5a    | \Conefilewithoptions@clashchk: Separated out from \Conefilewithoptions . . . . .                                                | 1003     |
|                                 | \__hook_patch_retokenize:Nnnn: Make patching of commands a global operation (gh/674) . . . . .        | 325 | 2021-11-30 ltexpl.dtx v1.3d     | \skipeval: Moved over from xfp package (gh/711) . . . . .                                                                       | 77       |
| 2021-09-28 lthooks.dtx v1.0s    | \__hook_if_usable_use:n: Correct usage of older \@@_if_file_hook:wTF (gh/675) . . . . .               | 286 | 2021-12-02 ltemd.dtx v1.0i      | \__cmd_run_code:: Correct defaults for optional arguments in nNNnnend-of-environment code (gh/712) . . . . .                    | 123      |
|                                 | \__hook_try_declarngeneric_hook_split: Correct usage of older \@@_if_file_hook:wTF (gh/675) . . . . . | 244 |                                 | \__cmd_start_aux:ccnnnn: Correct defaults for optional arguments in end-of-environment code (gh/712) . . . . .                  | 123      |
| 2021-10-14 ltboxes.dtx v1.4c    | \@mpfootnotetext: Explicitly set \@currentcounter (gh/687) . . . . .                                  | 832 | 2021-12-07 ltexpl.dtx v1.3d     | \skipeval: Added \dimeval and \skipeval (gh/711) . . . . .                                                                      | 78       |
| 2021-10-14 ltfiles.dtx v1.2p    | \@include: Warn about use in preamble . . . . .                                                       | 452 | 2021-12-11 ltdirchk.dtx v1.3a   | General: Add comment lines into latex.ltx to indicate temp definitions that are later overwritten (gh/725) . . . . .            | 2        |
|                                 | \@footnotetext: Explicitly set \@currentcounter (gh/687) . . . . .                                    | 930 | 2021-12-12 ltoutenc.dtx v2.0y   | General: \DeclareUnicodeAccent now makes the encoding argument implicit (gh/253) . . . . .                                      | 500      |
| 2021-10-14 ltmath.dtx v1.2j     | \@eqnset: Explicitly set \@currentcounter (gh/687) . . . . .                                          | 795 |                                 | Added \DeclareUnicodeCommand and \DeclareUnicodeSymbol (gh/253) . . . . .                                                       | 500      |
|                                 | \eqnarray: Explicitly set \@currentcounter (gh/687) . . . . .                                         | 799 | 2021-12-27 ltluatex.dtx v1.1x   | \newprotectedluacmd: Macros added . . . . .                                                                                     | 51       |
| 2021-10-15 ltfsdcl.dtx v3.0y    | \DeclareMathVersion: Initialize variable for freezing math version (gh/676) . . . . .                 | 613 | 2021-12-28 ltexpl.dtx v1.3d     | \ExpandArgs: Added document level names for \exp_args:Nc and the like (gh/735) . . . . .                                        | 78       |
|                                 | \newprotectedluacmd: Macros added . . . . .                                                           | 51  | 2021-13-31 ltemd.dtx v1.0j      | \__cmd_show:x: Make \ShowCommand stop for interaction . . . . .                                                                 | 152, 153 |
| 2021-10-15 ltluatex.dtx v1.1v   | General: provide_ charproc_ data added . . . . .                                                      | 63  |                                 |                                                                                                                                 |          |
| 2021-10-16 ltoutenc.dtx v2.0x   | \add@accent: Dont set \spacefactor in math mode gh/643 . . . . .                                      | 473 |                                 |                                                                                                                                 |          |

|                                                                                                                                 |      |                                                                                                                       |          |
|---------------------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------|----------|
| 2022-01-06 ltexpl.dtx v1.3e                                                                                                     |      | 2022-02-21 ltkeys.dtx v1.0e                                                                                           |          |
| \ExpandArgs: Adjust document level<br>names for \exp_ args:Nc and the<br>like (gh/735) . . . . .                                | 78   | \__keys_options_expand_module:Nn:<br>Faster approach to module<br>expansion . . . . .                                 | 1033     |
| 2022-01-06 ltshipout.dtx v1.0l                                                                                                  |      | 2022-02-28 ltcmd.dtx v1.0k                                                                                            |          |
| \AtBeginShipoutNext: Correctly<br>simulate \AtBeginShipout and<br>\AtBeginShipoutNext without<br>extending the syntax . . . . . | 1085 | General: Move latexrelease<br>redefinitions from ltcmd.dtx . . . . .                                                  | 193      |
| \AtEndDvi: Correctly simulate<br>\AtEndDvi without extending the<br>syntax . . . . .                                            | 1084 | 2022-02-28 ltexpl.dtx v1.3f                                                                                           |          |
| 2022-01-15 ltkeys.dtx v1.0b                                                                                                     |      | General: Move latexrelease<br>redefinitions from ltcmd.dtx . . . . .                                                  | 76       |
| \__keys_options_aux:n: Clear option<br>list in end-of-package hook . . . . .                                                    | 1030 | 2022-02-28 ltvers.dtx v1.1i                                                                                           |          |
| 2022-01-25 ltpplain.dtx v2.3h                                                                                                   |      | \new@moduledate: Detect a missing<br>\IncludeInRelease{0000/00/00}. . . . .                                           | 40       |
| \obeyspaces: Provide redirection to<br>support special use cases (gh/367) . . . . .                                             | 28   | 2022-03-10 ltbibl.dtx v1.1t                                                                                           |          |
| 2022-02-05 ltkeys.dtx v1.0b                                                                                                     |      | \cite: Ensure that an empty<br>argument generates a warning<br>(gh/790) . . . . .                                     | 938      |
| .usage: Create properties in<br>ltlkeys . . . . .                                                                               | 1028 | 2022-03-10 ltfilehook.dtx v1.0m                                                                                       |          |
| 2022-02-07 ltkeys.dtx v1.0c                                                                                                     |      | @\curr@file@reqd: Add<br>\set@curr@file@nosearch for<br>graphicx . . . . .                                            | 1049     |
| .usage: Correct ..code property . . . . .                                                                                       | 1028 | 2022-03-18 ltclass.dtx v1.5b                                                                                          |          |
| 2022-02-15 ltkeys.dtx v1.0c                                                                                                     |      | \load@onefilewithoptions: Switch to<br>\ProcessKeyOptions . . . . .                                                   | 1002     |
| \DeclareKeys: Expand module<br>argument . . . . .                                                                               | 1033 | 2022-03-18 ltcmd.dtx v1.0l                                                                                            |          |
| \DeclareUnknownKeyHandler: Added<br>\DeclareUnknownKeysHandler . . . . .                                                        | 1033 | \__cmd_cmd_type_cases:NnnnnnF: Fix<br>\@@_cmd_type_cases:NnnnnF<br>prematurely expanding macros<br>(gh/795) . . . . . | 181      |
| \ProcessKeyOptions: Expand module<br>argument . . . . .                                                                         | 1033 | 2022-03-18 ltkeys.dtx v1.0f                                                                                           |          |
| \SetKeys: Expand module argument . . . . .                                                                                      | 1035 | \__keys_options:n: Simplify to<br>always cover global options . . . . .                                               | 1030     |
| 2022-02-16 ltkeys.dtx v1.0d                                                                                                     |      | \__keys_options_global:n: Simplify<br>to always cover global options . . . . .                                        | 1030     |
| \DeclareKeys: Allow for active<br>characters in module argument . . . . .                                                       | 1033 | \ProcessKeyOptions: Remove<br>\ProcessKeyPackageOptions . . . . .                                                     | 1033     |
| \DeclareUnknownKeyHandler: Allow<br>for active characters in module<br>argument . . . . .                                       | 1033 | 2022-04-01 ltfiles.dtx v1.2q                                                                                          |          |
| Rename<br>\DeclareUnknownKeysHandler to<br>\DeclareUnknownKeyHandler . . . . .                                                  | 1033 | @\include: Process some hooks is an<br>include file is bypassed . . . . .                                             | 453      |
| \ProcessKeyOptions: Allow for active<br>characters in module argument . . . . .                                                 | 1033 | 2022-04-01 lthooks.dtx v1.0t                                                                                          |          |
| \SetKeys: Allow for active characters<br>in module argument . . . . .                                                           | 1035 | \c_hook_generic_include/.end_tl:<br>Support generic<br>include/.../excluded hooks . . . . .                           | 250      |
| 2022-02-19 ltcmd.dtx v1.0k                                                                                                      |      | 2022-04-03 ltfinal.dtx v2.2s                                                                                          |          |
| \IfBlankTF: Added \IfBlankTF and<br>friends . . . . .                                                                           | 192  | General: Integration of new mark<br>management interface . . . . .                                                    | 1174     |
| 2022-02-20 ltfinal.dtx v2.2r                                                                                                    |      | 2022-04-03 ltoutput.dtx v1.ih                                                                                         |          |
| \@uclclist: Use<br>\@expl@text@uppercase@n,<br>removing local redefinition of<br>\UTF@two@octets@noexpand . . . . .             | 1188 | \opcol: Interface with new mark<br>mechanism . . . . .                                                                | 1110     |
| use \text_lowercase:n . . . . .                                                                                                 | 1188 | 2022-04-04 ltpage.dtx v1.0n                                                                                           |          |
|                                                                                                                                 |      | \markright: Interface with new mark<br>mechanism . . . . .                                                            | 970, 971 |
|                                                                                                                                 |      | 2022-04-08 ltmath.dtx v1.2k                                                                                           |          |
|                                                                                                                                 |      | \openup: Make \protected (gh/123) . . . . .                                                                           | 790      |

|                                                                                                                                                             |                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2022-04-12 ltxref.dtx LaTeXe<br>\pageref: Macro reimplemented with<br>a starred version ..... 740                                                           | 2022-06-15 lthooks.dtx v1.0v<br>\__hook_activate_generic:n: Ensure<br>that a newly activated generic<br>hook gets its execution code set . 234                                          |
| \ref: Macro reimplemented with a<br>starred version ..... 740                                                                                               |                                                                                                                                                                                         |
| 2022-04-12 ltxref.dtx v1.1p<br>General: Add starred variants for the<br>ref commands ..... 738                                                              | \__keys_options_class:n: Better<br>handling of option removal ... 1031                                                                                                                  |
| \Ref: Macro reimplemented with a<br>starred version ..... 744                                                                                               | \__keys_options_package:n: Better<br>handling of option removal ... 1031                                                                                                                |
| 2022-04-21 ltfinal.dtx v2.2t<br>\@uclclist: Support \noexpand in<br>argument of<br>\@expl@text@uppercase@@n ... 1188                                        | 2022-06-16 ltkeys.dtx v1.0g<br>\__keys_options_class:n: Further<br>work on handling of option<br>removal ..... 1031                                                                     |
| 2022-05-06 ltmarks.dtx v1.0c<br>\__mark_new_class:nn: Wrong<br>command made \conlypreamble 950                                                              | \__keys_options_package:n: Further<br>work on handling of option<br>removal ..... 1031                                                                                                  |
| 2022-05-08 ltmath.dtx v1.2l<br>General: Use consistent math styles<br>under LuaTeX ..... 797                                                                | \__keys_options_package:nnn: New<br>function ..... 1031                                                                                                                                 |
| 2022-05-08 ltshipout.dtx v1.0m<br>\@kernel@after@enddocument@afterlastpage<br>Handle case where<br>shipout/lastpage is run too early<br>(gh/813) ..... 1081 | 2022-06-20 ltclass.dtx v1.5c<br>\load@onefilewithoptions: Pass raw<br>options to \ProcessKeyOptions 1002                                                                                |
| \__shipout_execute_main_cont:Nnnn:<br>Handle case where<br>shipout/lastpage is run too early<br>(gh/813) ..... 1071                                         | 2022-06-20 ltkeys.dtx v1.0h<br>\__keys_options_class:n: Use raw<br>options data ..... 1031                                                                                              |
| 2022-05-13 lthooks.dtx v1.0u<br>\__hook_use_once_clear:n: Check if<br>prop exists to avoid l3debug error 288                                                | \__keys_options_class:nnn: New<br>function ..... 1031                                                                                                                                   |
| 2022-05-17 lthooks.dtx v1.0u<br>\__hook_initialize_hook_code:n:<br>Refuse sorting one-time hooks<br>(gh/818). ..... 265                                     | \__keys_options_global:n: Use raw<br>options data ..... 1030                                                                                                                            |
| 2022-05-17 ltmeta.dtx v1.0b<br>General: Default definition for targets<br>added ..... 419                                                                   | \__keys_options_local:: Use raw<br>options data ..... 1032                                                                                                                              |
| 2022-05-27 ltfiles.dtx v1.2r<br>\listfiles: Try saved version string,<br>if ver@.. is \relax (gh/825) ... 463                                               | \__keys_options_package:n: Use raw<br>options data ..... 1031                                                                                                                           |
| 2022-05-27 ltoutenc.dtx v2.0z<br>General: Save the version string<br>(gh/825) ..... 511                                                                     | 2022-06-22 ltkeys.dtx v1.0i<br>.usage: Add ..notif property .... 1028                                                                                                                   |
| 2022-06-01 ltmarks.dtx v1.0d<br>\__mark_prepare_and_extract:nn:<br>Extend the logic for detecting the<br>marks in the box (gh/836) ..... 953                | 2022-06-30 ltfinal.dtx v2.2u<br>\@uclclist: Add<br>\AddToNoCaseChangeList .... 1189                                                                                                     |
| General: Marks are kernel errors ... 962                                                                                                                    | 2022-06-30 ltfinal.dtx v2.2v<br>\@uclclist: Just use<br>\text_lowercase:n without<br>\protectd@edf gh/881x .... 1188                                                                    |
| 2022-06-02 ltfinal.dtx v2.2u<br>\@uclclist: Add \NoCaseChange .. 1189                                                                                       | 2022-07-04 ltfinal.dtx v2.2w<br>\@uclclist: Introduced \CaseSwitch,<br>\DeclareCaseChangeEquivalent<br>and \MakeTitlecase to support<br>hooking into case changing<br>gh/889 ..... 1188 |
|                                                                                                                                                             | 2022-07-04 ltfssbas.dtx v3.2k<br>\frozen@everydisplay: Ignore spaces<br>if necessary (gh/886) ..... 535                                                                                 |
|                                                                                                                                                             | \frozen@everymath: Ignore spaces if<br>necessary (gh/886) ..... 535                                                                                                                     |

|                                                                                                                                                |      |                                                                                      |          |
|------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------|----------|
| 2022-07-04 ltfssdcl.dtx v3.0z                                                                                                                  |      | 2022-09-07 ltboxes.dtx v1.4d                                                         |          |
| \freeze@math@version: Ignore spaces<br>if necessary (gh/886) . . . . .                                                                         | 611  | \@iiminiplate: Check for nested<br>minipages and warn (gh/168) . . . . .             | 831      |
| 2022-07-05 ltkeys.dtx v1.0i                                                                                                                    |      | \if@in@minipage@env: Check for<br>nested minipages and warn<br>(gh/168) . . . . .    | 831      |
| \__keys_options_aux:n: Support<br>\CurrentOption . . . . .                                                                                     | 1030 | 2022-09-17 ltfssdcl.dtx v3.1a                                                        |          |
| \__keys_options_class:nnn: Correct<br>naming of raw class options<br>storage . . . . .                                                         | 1031 | \DeclareMathVersion: New logic for<br>freezing math versions (gh/921) . . . . .      | 613      |
| 2022-07-23 ltkeys.dtx v1.0j                                                                                                                    |      | \freeze@math@version: New logic for<br>freezing math versions (gh/921) . . . . .     | 610      |
| \__keys_options_end:: Output<br>'public' package name in<br>messages . . . . .                                                                 | 1030 | 2022-09-20 ltfssdcl.dtx v3.1b                                                        |          |
| \__keys_options_loaded:nn: Output<br>'public' package name in<br>messages . . . . .                                                            | 1034 | \DeclareSymbolFont: Drop any<br>surplus 'm' in series argument<br>(gh/918) . . . . . | 614      |
| 2022-08-07 lttextcomp.dtx v1.0g                                                                                                                |      | \SetSymbolFont: Drop any surplus 'm'<br>in series argument (gh/918) . . . . .        | 616      |
| \DeclareEncodingSubset: Make<br>global declaration (gh/905) . . . . .                                                                          | 697  | 2022-10-03 ltluatex.dtx v1.2a                                                        |          |
| 2022-08-10 ltcmd.dtx v1.1a                                                                                                                     |      | General: Add rules for callback<br>ordering . . . . .                                | 60       |
| \__cmd_arg_to_keyvalue:nn: New<br>internal arg-to-keyval processor . . . . .                                                                   | 177  | add_to_callback: Add rules for<br>callback ordering . . . . .                        | 68       |
| \__cmd_grab_D_long_obey_spaces_no_strip:w:<br>Add support for skipping brace<br>stripping . . . . .                                            | 157  | declare_callback_rule: Add<br>function . . . . .                                     | 69       |
| \__cmd_grab_R_aux>NNnN: Track<br>changes in D-type implementation . . . . .                                                                    | 163  | remove_from_callback: Add rules for<br>callback ordering . . . . .                   | 70       |
| \__cmd_grab_b_end:Nw: Track changes<br>in D-type implementation . . . . .                                                                      | 156  | 2022-10-10 ltclass.dtx v1.5d                                                         |          |
| General: New switch . . . . .                                                                                                                  | 115  | \@unprocessedoptions: Use<br>\protected@edef. . . . .                                | 1009     |
| 2022-08-13 ltluatex.dtx v1.1y                                                                                                                  |      | \load@onefilewithoptions: Use<br>\protected@edef. . . . .                            | 1002     |
| General: shared_callbacks added . . . . .                                                                                                      | 64   | \PassOptionsToClass: Use<br>\protected@xdef. . . . .                                 | 990      |
| add_to_callback: Adapted code for<br>shared_callbacks . . . . .                                                                                | 68   | \ProcessOptions*: Use<br>\protected@edef. . . . .                                    | 992      |
| remove_from_callback: Adapted code<br>for shared_callbacks . . . . .                                                                           | 70   | 2022-10-20 ltclass.dtx v1.5e                                                         |          |
| \mlist_to_hlist: Use shared_callback<br>system for<br>pre/post/mlist_to_hlist . . . . .                                                        | 72   | \load@onefilewithoptions: Define<br>key option handler in ltkeys . . . . .           | 1002     |
| 2022-08-18 ltfilehook.dtx v1.0n                                                                                                                |      | 2022-10-20 ltkeys.dtx v1.0l                                                          |          |
| \@disable@packageload@do: Inhibit<br>checking the loaded version when<br>package is load-disabled, and write<br>to the .log (gh/888) . . . . . | 1055 | \__keys_options_aux:n: Define key<br>option handler in ltkeys . . . . .              | 1030     |
| 2022-08-20 ltoutput.dtx v1.4j                                                                                                                  |      | \__keys_options_loaded:nn: Correct<br>an argument for a message . . . . .            | 1034     |
| \stockheight: Register added . . . . .                                                                                                         | 1099 | 2022-10-22 ltclass.dtx v1.5e                                                         |          |
| \stockwidth: Register added . . . . .                                                                                                          | 1099 | \@fileswithoptions: Use<br>\protected@xdef. . . . .                                  | 999      |
| 2022-08-21 ltkeys.dtx v1.0k                                                                                                                    |      | \@use@option: Use \detokenize . . . . .                                              | 994      |
| \__keys_options_loaded:nn: Correct<br>error message . . . . .                                                                                  | 1034 | \ProcessOptions*: Use<br>\detokenize . . . . .                                       | 992, 993 |
| 2022-09-03 ltmath.dtx v1.2m                                                                                                                    |      | 2022-10-22 ltdefns.dtx v1.5r                                                         |          |
| \smash: Guard against reboxing in<br>fractions (gh/517) . . . . .                                                                              | 789  | \@expandtwoargs: Use<br>\protected@edef. . . . .                                     | 89       |

|                                |                                                                                                                                                                               |           |                                 |                                                                                                                                                       |      |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 2022-10-22 ltkeys.dtx v1.01    | <code>\__keys_options_class:nnn:</code> Correct handling of unused option list .                                                                                              | 1031      | 2023-01-19 ltluatex.dtx v1.2b   | General: Remove unused local variable <code>tex_setattribute</code> . . . . .                                                                         | 54   |
| 2022-10-26 ltfinal.dtx v2.2x   | <code>@uclclist:</code> Auto-detect <code>babel</code> locale . . . . .                                                                                                       | 1188      | 2023-01-30 ltpara.dtx v1.01     | <code>\g__para_standard_everypar_tl:</code> Backout <code>\parskip</code> at top of <code>minipage</code> (gh/989) . . . . .                          | 410  |
|                                | Introduce optional argument for case-changing commands . . . . .                                                                                                              | 1188      | 2023-03-12 ltcmd.dtx v1.1c      | <code>\__cmd_copy_expandable:NnNNNnnn:</code> Distinguish (non-expandable) document commands starting with <code>\@@_start_expandable:nNNNNn</code> . | 146  |
|                                | Make case changing commands language-aware . . . . .                                                                                                                          | 1188      |                                 | <code>\__cmd_show:x:</code> Distinguish (non-expandable) document commands starting with <code>\@@_start_expandable:nNNNNn</code> .                   | 151  |
| 2022-11-06 ltmiscen.dtx v1.2a  | <code>\enddocument:</code> Repeat release info at the end (gh/944) . . . . .                                                                                                  | 762       | 2023-03-22 ltspace.dtx v1.3p    | <code>\samepage:</code> Add <code>\predisplaypenalty</code> setting (gh/1022) . . . . .                                                               | 424  |
| 2022-11-06 ltvers.dtx v1.1j    | General: Repeat release info at the end (gh/944) . . . . .                                                                                                                    | 38        | 2023-03-28 ltclass.dtx v1.5g    | <code>\IfFormatAtLeastTF:</code> Added <code>\IfFileAtLeastTF</code> (gh/1015) . . . . .                                                              | 983  |
| 2022-11-08 ltshipout.dtx v1.0n | <code>\__shipout_execute_main_cont:Nnnn:</code> Add shipout hook (gh/920) . . . . .                                                                                           | 1071      | 2023-03-28 ltfinal.dtx v2.2z    | <code>@uclclist:</code> Use groups for gh/1021                                                                                                        | 1189 |
|                                | <code>shipout/lastpage:</code> Add shipout hook (gh/920) . . . . .                                                                                                            | 1073      | 2023-04-01 ltfssbas.dtx v3.2l   | <code>\math@version:</code> Reset frozen mathversion gh/1028 . . . . .                                                                                | 534  |
| 2022-11-16 ltclass.dtx v1.5f   | <code>\endfilecontents:</code> Do not show "current dir" in message (gh/917) . . . . .                                                                                        | 1012      | 2023-04-02 ltfilehook.dtx v1.0o | <code>\@set@curr@file@aux:</code> Make <code>\@set@curr@file@aux \long</code> gh/942 . . . . .                                                        | 1051 |
|                                | Introduce key 'nowarn' on <code>filecontents</code> (gh/958) . . . . .                                                                                                        | 1010–1012 | 2023-04-06 ltcmdhooks.dtx v1.0g | <code>\__hook_double_hashes_space:w:</code> Add case for <code>\c_@@_hashes_tl</code> (hook-args) . . . . .                                           | 319  |
| 2022-11-24 ltdefns.dtx v1.5s   | <code>\DeclareEnvironmentCopy:</code> Add <code>\NewEnvironmentCopy</code> , <code>\RenewEnvironmentCopy</code> , and <code>\DeclareEnvironmentCopy</code> (gh/963) . . . . . | 99        |                                 | <code>\__hook_make_prefixes:w:</code> Rename to <code>\c_@@_hashes_tl</code> (hook-args) . . . . .                                                    | 315  |
|                                | <code>\ShowEnvironment:</code> Added <code>\ShowEnvironment</code> . . . . .                                                                                                  | 104       |                                 | <code>\c__hook_hashes_tl:</code> Rename to <code>\c_@@_hashes_tl</code> and add <code>\c_@@_hash_tl</code> (hook-args) . . . . .                      | 309  |
| 2022-11-28 ltspace.dtx v1.3o   | <code>\@hspace:</code> Support calc syntax correctly (gh/967) . . . . .                                                                                                       | 439       | 2023-04-06 lthooks.dtx v1.1a    | <code>\__hook_activate_generic:n:</code> Changes to add hook arguments (hook-args) . . . . .                                                          | 234  |
|                                | <code>\@vspace@calcify:</code> Support calc syntax without a group (gh/967)                                                                                                   | 426       |                                 | <code>\__hook_braced_real_loop:w:</code> Macro added (hook-args) . . . . .                                                                            | 258  |
| 2022-11-29 ltcmd.dtx v1.1b     | <code>\__cmd_show:x:</code> Add <code>\@showenvironmentlisthook</code> . . . . .                                                                                              | 153       |                                 | <code>\__hook_chk_args_allowed:nn:</code> Macro added (hook-args) . . . . .                                                                           | 242  |
| 2022-11-30 ltcmd.dtx v1.1b     | <code>\__cmd_show:x:</code> Don't stop for the <code>\begin</code> part of an environment . . . . .                                                                           | 152, 153  |                                 | <code>\__hook_clear_next:n:</code> Changes to add hook arguments (hook-args) . . . . .                                                                | 283  |
|                                | <code>\begin</code> part of an environment . . . . .                                                                                                                          | 152, 153  |                                 | <code>\__hook_code_gset_aux:nnn:</code> Macro added (hook-args) . . . . .                                                                             | 254  |
| 2022-11-30 ltfinal.dtx v2.2y   | <code>@uclclist:</code> Set <code>\oe/\OE</code> equal to act as a marker for <code>babel</code> . . . . .                                                                    | 1189      |                                 | <code>\__hook_code_gset_auxi:een:</code> Macro added (hook-args) . . . . .                                                                            | 252  |
| 2023-01-05 ltfiles.dtx v1.2s   | <code>\document:</code> <code>\do</code> now with default definition in the kernel (gh/975)                                                                                   | 446       |                                 |                                                                                                                                                       |      |

|                                        |                                                                                       |          |
|----------------------------------------|---------------------------------------------------------------------------------------|----------|
| \_\_hook\_cs\_end:w:                   | Macro added<br>(hook-args).                                                           | 258      |
| \_\_hook\_cs\_if\_empty:c:             | Macro added<br>(hook-args).                                                           | 257      |
| \_\_hook_gput_next_do:nn:              | Changes to add hook arguments<br>(hook-args).                                         | 282      |
| \_\_hook_gremove_code:nn:              | Changes to add hook arguments<br>(hook-args).                                         | 251      |
| \_\_hook_if_execute_immediately:n:     | Changes to add hook arguments<br>(hook-args).                                         | 288      |
| \_\_hook_init_structure:n:             | Changes to add hook arguments<br>(hook-args).                                         | 230      |
| \_\_hook_initialize_all::              | Changes to add hook arguments<br>(hook-args).                                         | 263      |
| \_\_hook_initialize_hook_code:n:       | Changes to add hook arguments<br>(hook-args).                                         | 265      |
| \_\_hook_initialize_single:ccn:        | Changes to add hook arguments<br>(hook-args).                                         | 268      |
| \_\_hook_log:nN:                       | Changes to add hook arguments (hook-args).                                            | 275      |
| \_\_hook_log_next_code:n:              | Changes to add hook arguments<br>(hook-args).                                         | 279      |
| \_\_hook_make_usable:nn:               | Changes to add hook arguments (hook-args).                                            | 228      |
| \_\_hook_new:nn:                       | Add<br>\hook_new_with_args:nn<br>(hook-args).                                         | 227      |
|                                        | Update hook code after declaring.                                                     | 228      |
| \_\_hook_new_reversed:nn:              | Add<br>\hook_new_reversed_with_-<br>args:nn<br>(hook-args).                           | 231      |
| \_\_hook_normalise_cs_args:nn:         | Macro added (hook-args).                                                              | 254      |
| \_\_hook_parameter:n:                  | Macro added<br>(hook-args).                                                           | 259      |
| \_\_hook_post_initialization_defs::    | Macro added (hook-args).                                                              | 285      |
| \_\_hook_prop_gput_labeled_do:Nnnn:    | Add<br>\hook_gput_code_with_args:nnn<br>(hook-args).                                  | 239      |
|                                        | Changes to add hook arguments<br>(hook-args).                                         | 239, 240 |
|                                        | Macro added (hook-args).                                                              | 241      |
| \_\_hook_set_normalise_fn:nn:          | Macro added (hook-args).                                                              | 255      |
| \_\_hook_t1_set:cn:                    | Clean-up unused<br>variants (hook-args).                                              | 224      |
| \_\_hook_try_declarng_generic_hook:wn: | Changes to add hook arguments<br>(hook-args).                                         | 245      |
| \_\_hook_use_i_delimit_by_s_mark:nw:   | Use standard naming scheme<br>(hook-args).                                            | 224      |
| \_\_hook_use_initialized:nnw:          | Add<br>\hook_use:nnw (hook-args).                                                     | 285      |
| \_\_hook_use_once:nn:                  | Changes to<br>add hook arguments (hook-args).                                         | 287      |
| \_\_hook_use_once_clear:n:             | Changes to add hook arguments<br>(hook-args).                                         | 288      |
| General:                               | Add dedicated rollback code<br>to revert data structures<br>(hook-args).              | 302      |
|                                        | Messages 'too-many-args',<br>'without-args' and 'one-time-args'<br>added (hook-args). | 293      |
| \AddToHookNextWithArguments:           | Add<br>\AddToHookNextWithArguments<br>(hook-args).                                    | 297      |
| \AddToHookWithArguments:               | Add<br>\AddToHookWithArguments<br>(hook-args).                                        | 296      |
| \c\_hook_??_parameter_t1:              | Token<br>list added (hook-args).                                                      | 260      |
| \c\_hook_nine_parameters_t1:           | Add<br>auxiliary token lists (hook-args).                                             | 224      |
| \g\_hook_replacing_stack_seq:          | Macro added (hook-args).                                                              | 292      |
| \hook_gput_next_code:nn:               | Add<br>\hook_gput_next_code_with_-<br>args:nn<br>(hook-args).                         | 281      |
| \hook_if_empty:n:                      | Changes to add<br>hook arguments (hook-args).                                         | 289      |
| \hook_new_pair_with_args:nnn:          | Add<br>\hook_new_pair_with_args:nnn<br>(hook-args).                                   | 231      |
| \hook_use_once:nnw:                    | Add<br>\hook_use_once:nnw (hook-args).                                                | 286      |
| \NewMirroredHookPairWithArguments:     | Add \NewHookWithArguments<br>(hook-args).                                             | 296      |
| \UseOneTimeHookWithArguments:          | Add<br>\UseHookWithArguments<br>(hook-args).                                          | 298      |

|                                 |                                                                                                                                                                          |      |                                                                                                                                 |                                                                                                                         |      |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|------|
| 2023-04-11 ltfinal.dtx v2.3a    | <code>\@uclclist</code> : Use new generic mechanism to detect locale . . . . .                                                                                           | 1188 | 2023-05-21 lthooks.dtx v1.1d                                                                                                    | <code>\__hook_if_cmd_hook:w</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .                | 291  |
| 2023-04-13 ltcmd.dtx v1.1d      | <code>\__cmd_grab_v_group_end::</code> : Set <code>\tex_endlinechar:D</code> earlier (gh/876). . . . .                                                                   | 164  | <code>\__hook_make_usable:nn</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .                       | 229                                                                                                                     |      |
| 2023-04-14 ltclass.dtx v1.5h    | <code>\load@onefilewithoptions</code> : Define <code>\load@onefilewithoptions</code> when in <code>latexrelease</code> (gh/992) . . . . .                                | 1001 | <code>\__hook_new:nn</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .                               | 228                                                                                                                     |      |
| 2023-04-15 ltplain.dtx v2.3i    | <code>\extrafloats</code> : Protect box 255 in lualatex gh/1041 . . . . .                                                                                                | 21   | <code>\__hook_try_declar ing_generic_hook:wn:</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .      | 245                                                                                                                     |      |
| 2023-04-16 ltfinal.dtx v2.3a    | <code>\BCPdata</code> : Command added . . . . .                                                                                                                          | 1187 | <code>\__hook_try_declar ing_generic_next_hook:nn:</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . . | 243                                                                                                                     |      |
| 2023-04-16 lthooks.dtx v1.1b    | <code>\__hook_include_legacy_code_chunk:n</code> :<br><code>\@@replacing_args_false</code> : in<br><code>\@@include_legacy_code_- chunk:n</code> . . . . .               | 232  | <code>\c_hook_parameter_cmd./after_tl:</code> : Token lists added (cmd-args). . . . .                                           | 250                                                                                                                     |      |
| 2023-04-19 ltfinal.dtx v2.3b    | <code>\@uclclist</code> : Add<br><code>\DeclareLowercaseMapping</code> ,<br><code>\DeclareTitlecaseMapping</code> and<br><code>\DeclareUppercaseMapping</code> . . . . . | 1189 | 2023-05-26 ltcmd.dtx v1.1e                                                                                                      | <code>\__cmd_defaults_error:w</code> : Use simpler variant <code>\cs_generate_- from_arg_count&gt;NNno</code> . . . . . | 124  |
| 2023-04-19 lthooks.dtx v1.1c    | <code>\__hook_gput_next_do:nn</code> : Initialise hook structure when adding 'next' code (gh/1052). . . . .                                                              | 282  | 2023-05-30 ltfinal.dtx v2.3c                                                                                                    | <code>\@uclclist</code> : Fix a typo in implementation of <code>\DeclareLowercaseMapping</code> , etc. . . . .          | 1189 |
|                                 | <code>\hook_if_empty:n</code> : Simpler and faster version (gh/1052). . . . .                                                                                            | 289  | 2023-06-06 lthooks.dtx v1.1e                                                                                                    | <code>\__hook_code_gset_auxi:een</code> : Short-circuit when the hook is declared without args (gh1078). . . . .        | 252  |
| 2023-05-12 ltxref.dtx v1.1q     | <code>\label</code> : (UFi)added a hook with argument . . . . .                                                                                                          | 742  | 2023-06-14 ltmiscen.dtx v1.2b                                                                                                   | <code>\vobeytabs</code> : Macro added . . . . .                                                                         | 777  |
|                                 | (UFi)extended to store five arguments . . . . .                                                                                                                          | 742  |                                                                                                                                 | <code>\verbatim*</code> : Support visible tabs in <code>\verb*</code> . . . . .                                         | 779  |
| 2023-05-13 ltmath.dtx v1.2n     | <code>\leqno</code> : add <code>\ignorespaces</code> gh/1059                                                                                                             | 797  | 2023-06-14 ltspage.dtx v1.3q                                                                                                    | <code>\xobeytab</code> : Macro added . . . . .                                                                          | 438  |
| 2023-05-15 ltfiles.dtx v1.2t    | <code>\IfFileExists@</code> : Use expl3 file existence test . . . . .                                                                                                    | 457  | 2023-06-15 ltmiscen.dtx v1.2b                                                                                                   | <code>\@@sverb</code> : Support visible tabs . . . . .                                                                  | 781  |
|                                 | <code>\IfFileExists@@</code> : Macro added . . . . .                                                                                                                     | 457  |                                                                                                                                 | <code>\setupverbvisiblespace</code> : Support visible tabs . . . . .                                                    | 780  |
| 2023-05-21 ltcmdhooks.dtx v1.0h | <code>\__hook_guess_arg_count:nw</code> : Macro added (cmd-args). . . . .                                                                                                | 323  |                                                                                                                                 | <code>\vobleyspaces</code> : Support tabs . . . . .                                                                     | 777  |
|                                 | <code>\__hook_make_prefixes:w</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .                                                               | 316  | 2023-06-15 ltmiscen.dtx v1.2q                                                                                                   | <code>\setupverbvisibletab</code> : Provide visible tab in <code>\verb*</code> . . . . .                                | 781  |
|                                 | <code>\__hook_patch_retokenize:Nnnn</code> : Changes to allow support arguments in cmd hooks (cmd-args). . . . .                                                         | 324  | 2023-06-16 ltfiles.dtx v1.2u                                                                                                    | <code>\IfFileExists@@</code> : Support piped input . . . . .                                                            | 457  |
|                                 |                                                                                                                                                                          |      | 2023-07-02 lltuaretex.dtx v1.2c                                                                                                 | <code>new_luafunction</code> : Ensure existing table entries not overwritten                                            |      |
|                                 |                                                                                                                                                                          |      |                                                                                                                                 | gh/1100 . . . . .                                                                                                       | 59   |

|                                   |                                                                                                                                |     |                                                                                            |                                                                                                                                          |      |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------|-----|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|------|
| 2023-08-03 lltuatem.dtx v1.2c     | <code>\_list_to_hlist</code> : Fix callback type of <code>post_mlist_to_hlist_callback</code> . . . . .                        | 72  | 2023-12-01 ltcmd.dtx v1.2b                                                                 | <code>\_cmd_cmd_type_cases:NnnnnnF</code> : Extend for optimized commands . . . . .                                                      | 181  |
| 2023-08-19 ltcmd.dtx v1.2a        | General: Removed commands that should have remained only in <code>xparse.dtx</code> . . . . .                                  | 113 | <code>\_cmd_start_optimized::</code> : Optimize cmd creation for all-m arguments . . . . . | 118                                                                                                                                      |      |
| 2023-09-01 ltmiscen.dtx v1.2c     | <code>\@obeytabs</code> : Provide global definition for active tab . . . . .                                                   | 777 | 2023-12-22 ltcmd.dtx v1.2c                                                                 | General: Generalize message <code>invalid-bang</code> (gh/1198) . . . . .                                                                | 183  |
| 2023-09-06 ltmiscen.dtx v1.2b     | <code>\enddocument</code> : Test changes of values in <code>\new@label@record</code> . . . . .                                 | 762 | 2024-01-03 lthooks.dtx v1.1g                                                               | <code>\_hook_log:nN</code> : Fix expansion of <code>\@@_print_args:nn</code> argument (gh/1221) . . . . .                                | 275  |
| 2023-09-20 ltproperties.dtx v1.0c | <code>\RecordProperties</code> : use <code>\protected@edef</code> for safer handling of active chars. . . . .                  | 753 | 2024-01-13 ltkeys.dtx v1.0m                                                                | <code>\_keys_options_class:n</code> : Trim spaces off key names . . . . .                                                                | 1031 |
|                                   | <code>\SetProperty</code> : use <code>\protected@edef</code> . . . . .                                                         | 752 |                                                                                            | <code>\_keys_options_package:n</code> : Trim spaces off key names . . . . .                                                              | 1031 |
| 2023-10-13 ltexpl.dtx v1.3g       | <code>\IfExplAtLeastTF</code> : Provide a test for <code>expl3</code> date (gh/1004) . . . . .                                 | 79  | 2024-01-17 ltproperties.dtx v1.0d                                                          | <code>\_property_record_value_aux:e</code> : Use <code>\protected@write</code> . . . . .                                                 | 752  |
| 2023-10-26 ltboxes.dtx v1.4e      | <code>\@iframebox</code> : Guard against unknown alignment gh/1072 . . . . .                                                   | 826 | 2024-01-24 lthooks.dtx v1.1h                                                               | <code>\_hook_if_usable_use:n</code> : Correct usage of older <code>\@@_if_file_hook:wTF</code> (gh/1243) . . . . .                       | 286  |
|                                   | <code>\@imakebox</code> : Guard against unknown alignment gh/1072 . . . . .                                                    | 821 |                                                                                            | <code>\_hook_try_declarng_generic_hook_split:nNNnn</code> : Correct usage of older <code>\@@_if_file_hook:wTF</code> (gh/1243) . . . . . | 244  |
|                                   | <code>\@parboxto</code> : Guard against unknown alignment gh/1072 . . . . .                                                    | 828 | 2024-01-27 lttextcomp.dtx v1.1a                                                            | General: Added check file for encoding subset . . . . .                                                                                  | 727  |
| 2023-10-26 ltspace.dtx v1.3r      | <code>\nobreakspace</code> : Protected definition for tilde . . . . .                                                          | 437 |                                                                                            | Adjusted/corrected TS1 sub-encoding declarations for various families (gh/1257) . . . . .                                                | 707  |
| 2023-11-07 ltcounds.dtx v1.1n     | <code>\@definecounter</code> : Do not change <code>\the...</code> if already defined (gh/823) . . . . .                        | 514 | 2024-01-29 ltmarks.dtx v1.0e                                                               | <code>\_mark_extract_and_handle_marks:nn</code> : Macro added . . . . .                                                                  | 952  |
| 2023-11-07 ltoutenc.dtx v2.1a     | General: Add more explanation to error message (gh/1102) . . . . .                                                             | 510 |                                                                                            | <code>\_mark_update_structure_from_material:nn</code> : Macro renamed . . . . .                                                          | 955  |
| 2023-11-07 ltpplain.dtx v2.3j     | <code>\tracingnone</code> : Set <code>\tracinglostchars</code> to 2 in <code>\tracingnone</code> (gh/549) . . . . .            | 35  |                                                                                            | <code>\ShowMarksAt</code> : Macro added . . . . .                                                                                        | 964  |
| 2023-11-15 ltfiles.dtx v1.2v      | <code>\if@listfiles@hashes</code> : Extend file list information . . . . .                                                     | 463 | 2024-01-30 ltclass.dtx v1.5i                                                               | <code>\@fileswithoptions</code> : Test group level . . . . .                                                                             | 998  |
|                                   | <code>\if@listfiles@sizes</code> : Extend file list information . . . . .                                                      | 463 | 2024-03-18 ltthm.dtx v1.0g                                                                 | <code>\endtheorem</code> : Insert link target in the label (UFI) . . . . .                                                               | 897  |
|                                   | <code>\listfiles</code> : Extend file list information . . . . .                                                               | 463 |                                                                                            | <code>\@thm</code> : Use <code>\@kernel@refstepcounter</code> to avoid an unwanted target (UFI) . . . . .                                | 896  |
| 2023-11-16 ltpara.dtx v1.0m       | <code>\g__para_standard_everypar_tl</code> : Correct error message: hook left horizontal not vertical mode (gh/1182) . . . . . | 410 | 2024-03-21 ltcmd.dtx v1.2d                                                                 | <code>\_cmd_grab_v_aux_put:N</code> : Collect <code>\endlinechar</code> as <code>\obeyedline</code> . . . . .                            | 167  |
|                                   |                                                                                                                                |     | 2024-03-22 ltclass.dtx v1.5j                                                               | <code>\load@onefilewithoptions</code> : Apply one-step expansion to raw option                                                           |      |

|                                                                                                                                           |      |                                                                                                                                                                                                                               |      |
|-------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| list, to be consistent with change<br>for gh/580 (gh/1298). . . . .                                                                       | 1002 | regions (gh/1359) . . . . .                                                                                                                                                                                                   | 960  |
| 2024-04-10 ltclass.dtx v1.5k                                                                                                              |      | \_mark_update_dblcol_structures::<br>Correct logic for first mark in page<br>region if first column contains no<br>marks (gh/1359) . . . . .                                                                                  | 967  |
| \IfClassLoadedWithOptionsTF:<br>Provide T and F conditionals not<br>just TF (gh/1262) . . . . .                                           | 986  | \mark_insert:nn: Use sequence<br>marker to make all marks unique<br>on nearby regions (gh/1359) . . . . .                                                                                                                     | 960  |
| \IfFileLoadedF: Provide<br>\IfFileLoadedTF and variants<br>(gh/1222) . . . . .                                                            | 986  | insertmark: Use sequence marker to<br>make all marks unique on nearby<br>regions (gh/1359) . . . . .                                                                                                                          | 961  |
| 2024-04-17 ltcmd.dtx v1.2e                                                                                                                |      | 2024-05-31 ltmarks.dtx v1.0g                                                                                                                                                                                                  |      |
| \_\_cmd\_cmd\_type\_cases:NnnnnnF: Use<br>\_\_kernel\_cs\_parameter\_spec:N<br>instead of \cs_argument_-<br>spec:N/\cs_parameter_spec:N . | 181  | \_\_mark_new_class:nn: Initialize all<br>marks with an id, use 0 when a<br>new class is made (gh/1359) . . . . .                                                                                                              | 951  |
| 2024-04-17 ltcmdhooks.dtx v1.0j                                                                                                           |      | \mark_use_last:nn: Remove the id<br>when returning the mark value<br>(gh/1359) . . . . .                                                                                                                                      | 961  |
| General: Use<br>\_\_kernel\_cs\_parameter\_spec:N<br>instead of \cs_argument_-<br>spec:N/\cs_parameter_spec:N .                           | 308  | 2024-06-04 ltclass.dtx v1.5l                                                                                                                                                                                                  |      |
| 2024-04-17 ltdefns.dtx v1.5t                                                                                                              |      | @cls@pkg: Allow for extensions other<br>than "cls" and "pkg" . . . . .                                                                                                                                                        | 1008 |
| General: Rename<br>\@expl@cs@argument@spec@@N to<br>\@expl@cs@parameter@spec@@N<br>(gh/1014) . . . . .                                    | 80   | @onefilewithoptions: New<br>argument \pkgcls@ext (gh/870) . . . . .                                                                                                                                                           | 1019 |
| 2024-04-17 ltexpl.dtx v1.3h                                                                                                               |      | \pkgcls@parse@date@arg: New<br>argument \pkgcls@ext (gh/870) . . . . .                                                                                                                                                        | 1021 |
| General: Add a kernel-level copy of<br>\cs_parameter_spec:N . . . . .                                                                     | 77   | 2024-06-04 lthuataex.dtx v1.2d                                                                                                                                                                                                |      |
| Rename<br>\@expl@cs@argument@spec@@N to<br>\@expl@cs@parameter@spec@@N<br>(gh/1014) . . . . .                                             | 77   | provides_module: Add v to version<br>string if required (gh/1364) . . . . .                                                                                                                                                   | 54   |
| Update name of expl3 function . . . . .                                                                                                   | 77   | 2024-06-10 ltboxes.dtx v1.4g                                                                                                                                                                                                  |      |
| 2024-04-17 ltproperties.dtx v1.0e                                                                                                         |      | \finalstrut: Always use a \vrule<br>strut after all, but back up by a<br>baseline if already in vertical mode.<br>Otherwise empty table p-cells will<br>not get the correct width (bug seen<br>first with colortbl) . . . . . | 835  |
| \IfLabelExistsF: Renamed<br>\IfLabelExistTF to<br>\IfLabelExistsTF (gh/1262) . . . . .                                                    | 756  | 2024-06-10 lttagging.dtx v1.0b                                                                                                                                                                                                |      |
| \IfPropertyExistsF: Renamed<br>\IfPropertyExistTF to<br>\IfPropertyExistsTF (gh/1262) . . . . .                                           | 755  | \tbl_crcr:n: Always issue a \crcr<br>even if we are at the start of a row<br>to avoid problems with tabulary<br>and similar code . . . . .                                                                                    | 1171 |
| 2024-04-18 ltboxes.dtx v1.4f                                                                                                              |      | 2024-06-17 ltfssbas.dtx v3.2m                                                                                                                                                                                                 |      |
| \finalstrut: Use a \hrule strut not<br>a \vrule if already in vertical<br>mode (bug seen first with<br>footmisc/14) . . . . .             | 835  | \showhyphens: set \tracinglostchars<br>to 0 in \showhyphens . . . . .                                                                                                                                                         | 547  |
| 2024-04-22 lttextcomp.dtx v2.1a                                                                                                           |      | 2024-06-19 ltkeys.dtx v1.0n                                                                                                                                                                                                   |      |
| General: Drop default option info<br>(gh/1333) . . . . .                                                                                  | 716  | \keys_options_class:n: Refactor<br>function . . . . .                                                                                                                                                                         | 1031 |
| 2024-04-24 lttextcomp.dtx v2.1b                                                                                                           |      | \keys_options_class:nn: New<br>function . . . . .                                                                                                                                                                             | 1031 |
| General: Load the 2018 version when<br>rolling back prior to 2018-08-11<br>(gh/1333) . . . . .                                            | 716  | \keys_options_class:nnn:<br>Refactor function . . . . .                                                                                                                                                                       | 1031 |
| 2024-05-30 ltmarks.dtx v1.0f                                                                                                              |      | Track options used by classes . . . . .                                                                                                                                                                                       | 1031 |
| \_\_mark_id:n: Use sequence marker to<br>make all marks unique on nearby                                                                  |      | \keys_options_package:nn: New<br>functions . . . . .                                                                                                                                                                          | 1031 |

|                                                                                                                                                                    |      |                                                                                                                                                              |      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| \_\_keys\_options\_package:nnn: Skip options given to packages . . . . .                                                                                           | 1031 | 2024-07-13 ltagging.dtx v1.0c<br>tagsupport/tbl/leaders/end:<br>Sockets for \cline leaders added<br>(tagging/134) . . . . .                                  | 1165 |
| \_\_keys\_scope:N: New function . . . . .                                                                                                                          | 1028 | 2024-08-03 ltclass.dtx v1.5m<br>\pkcls@use@this@release: Add selected release to the file list<br>(gh/1413) . . . . .                                        | 1023 |
| \_\_keys\_scope:n: New function . . . . .                                                                                                                          | 1028 | 2024-08-09 lhooks.dtx v1.1i<br>\_hook_normalize_hook_args_aux:Nn:<br>Distinguish between empty label and empty hook (gh/1423) . . . . .                      | 236  |
| \1\_keys\_class\_only\_clist: New variable . . . . .                                                                                                               | 1029 | \_hook_parse_dot_label:nN:<br>Distinguish between empty label and empty hook (gh/1423) . . . . .                                                             | 235  |
| \1\_keys\_forced\_global\_clist: New variable . . . . .                                                                                                            | 1029 | \_hook_parse_label_default:nN:<br>Distinguish between empty label and empty hook (gh/1423) . . . . .                                                         | 235  |
| .pass-to-packages: New key property . . . . .                                                                                                                      | 1028 | General: New message “empty-hook” (gh/1423). . . . .                                                                                                         | 294  |
| 2024-06-20 ltkeys.dtx v1.0o<br>\_\_keys\_scope:N: Ensure only key name is stored . . . . .                                                                         | 1028 | 2024-08-16 lluatex.dtx v1.2e<br>provides_module: Support missing version string (gh/1443) . . . . .                                                          | 54   |
| 2024-06-23 ltboxes.dtx v1.4h<br>\color@endgroup: Adjust for new @endpe handling . . . . .                                                                          | 823  | 2024-09-03 fontdef.dtx v3.0k<br>General: Load also t1cmss.fd and t1cmmt.fd . . . . .                                                                         | 665  |
| 2024-06-23 llists.dtx v1.0u<br>\@doendpe: Set \if@endpe to false before calling \par (needed for tagging) . . . . .                                                | 812  | 2024-09-04 llength.dtx v1.1e<br>\@settodim: (UF) Suspend tagging . . . . .                                                                                   | 521  |
| \propagate@doendpe: Set \if@endpe globally and also set up migration to the outside . . . . .                                                                      | 813  | 2024-09-05 ltproperties.dtx v1.0f<br>\_property_record_value_aux:e:<br>Remove \if@filesw test to be in line with \label, tagging-project issue 696 . . . . . | 752  |
| 2024-06-23 ltmiscen.dtx v1.2d<br>\begin: Separate \begin and \end definitions for individual rollback . . . . .                                                    | 769  | 2024-09-10 ltspace.dtx v1.3s<br>\addpenalty: Drop unnecessary \noitemerr and instead generate \LRmoderr if we are in restricted hmode (gh/1460) . . . . .    | 433  |
| \end: Separate \begin and \end definitions for individual rollback . . . . .                                                                                       | 770  | \addvspace: Drop unnecessary \noitemerr and instead generate \LRmoderr if we are in restricted hmode (gh/1460) . . . . .                                     | 432  |
| \end\verbvisiblespace: Adjust for new @endpe handling . . . . .                                                                                                    | 772  | 2024-09-11 lterror.dtx v1.2u<br>\LRmoderr: Error message added . . . . .                                                                                     | 397  |
| 2024-06-23 ltpara.dtx v1.0n<br>\g\_para\_standard\_everypar\_tl:<br>Append \everypar toks to \g\_@_standard\_everypar\_tl, rollback 2023/06/01 (gh/1386) . . . . . | 411  | 2024-09-13 ltagging.dtx v1.0h<br>tagsupport/float/hmode/end:<br>Sockets for floats added . . . . .                                                           | 1166 |
| 2024-06-24 fontdef.dtx v3.0j<br>General: load ts1 cmr fd file in Unicode engines . . . . .                                                                         | 666  | 2024-09-20 ltcnts.dtx v1.1o<br>\addtoreset: add the parent theHfoo if a counter is reset . . . . .                                                           | 515  |
| 2024-07-06 ltcmd.dtx v1.2f<br>\cmd_declare_env:ennn: Use space-trimmed envname directly (gh/1399) . . . . .                                                        | 121  | \definecounter: define theHfoo (used for internal links) . . . . .                                                                                           | 514  |
| \NewDocumentEnvironment: Trim spaces from envname first (gh/1399) . . . . .                                                                                        | 189  |                                                                                                                                                              |      |
| 2024-07-10 ltmiscen.dtx v1.2e<br>General: Drop code chunks before adding them to avoid duplication in rollback (gh/1407) . . . . .                                 | 764  |                                                                                                                                                              |      |
| \enddocument: Drop code chunks before adding them to avoid duplication in rollback (gh/1407) . . . . .                                                             | 762  |                                                                                                                                                              |      |

|                                          |      |                                        |      |
|------------------------------------------|------|----------------------------------------|------|
| 2024-09-20 ltagging.dtx v1.0h            |      | 2024-10-21 ltagging.dtx v1.0k          |      |
| General: moved                           |      | \UseExpandableTaggingSocket:           |      |
| \@kernel@refstepcounter into             |      | Added expandable variants . . . . .    | 1162 |
| ltxref . . . . .                         | 1171 | Changed behavior of two argument       |      |
| 2024-09-20 ltxref.dtx v1.1r              |      | tagging sockets when disabled. . . . . | 1162 |
| \refstepcounter: add sockets . . . . .   | 742  | 2024-10-26 lcounts.dtx v1.1p           |      |
| provide a kernel copy                    |      | \Addtoreset: Fully expand counter      |      |
| kernel@refstepcounter . . . . .          | 742  | name in theHfoo commands               |      |
| set also currentHref . . . . .           | 742  | (gh/1508) . . . . .                    | 515  |
| 2024-09-21 ltmeta.dtx v1.0c              |      | 2024-10-27 ltsockets.dtx v0.9b         |      |
| General: Added tagging support . . . . . | 419  | \socket_set_plug:nnn: Make plug        |      |
| 2024-09-25 ltproperties.dtx v1.0g        |      | definition non-protected . . . . .     | 338  |
| \IfLabelExistsF: Fixed definitions of    |      | \socket_use_expandable:n: Added        |      |
| \IfLabelExistsT and                      |      | \socket_use_expandable:n . . . . .     | 340  |
| \IfLabelExistsF . . . . .                | 756  | 2024-10-29 lthooks.dtx v1.1k           |      |
| 2024-10-10 ltagging.dtx v1.0i            |      | \_hook_list_if_rule_exists:nmm:        |      |
| default: Restore also paratagging        |      | Skip mapping over undeclared           |      |
| (tagging/723) . . . . .                  | 1163 | \g_@@_{hook}_code_prop                 |      |
| 2024-10-12 ltmiscen.dtx v1.2f            |      | (gh/1513) . . . . .                    | 280  |
| \begin: Make \begin                      |      | \_hook_log:nN: Skip mapping over       |      |
| engine-protected . . . . .               | 769  | undeclared                             |      |
| 2024-10-21 lthooks.dtx v1.1j             |      | \g_@@_{hook}_code_prop                 |      |
| \IfHookEmptyF: Define                    |      | (gh/1513) . . . . .                    | 275  |
| \IfHookEmptyT, \IfHookEmptyF . . . . .   | 299  | v1.0b lttemplates.dtx 2024-02-15       |      |
| 2024-10-21 ltmarks.dtx v1.0h             |      | \IfInstanceExistsTF: New macros        | 380  |
| \IfMarksEqualF: Define                   |      | v1.0c lttemplates.dtx 2024-04-17       |      |
| \IfMarksEqualT, \IfMarksEqualF . . . . . | 964  | \IfInstanceExistsTF: Use plural        |      |
| 2024-10-21 ltproperties.dtx v1.0h        |      | names . . . . .                        | 380  |
| \IfPropertyRecordedF: Define             |      | v1.0d lttemplates.dtx 2024-10-07       |      |
| \IfPropertyRecordedT,                    |      | \_template_assign_variable::           |      |
| \IfPropertyRecordedF . . . . .           | 756  | Correct passing of \KeyValue           |      |
|                                          |      | contents . . . . .                     | 373  |

# 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           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \!                | 02:431, 02:433, 57:518                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| \!                | <u>38:214</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \"                | 04:512, 04:513, 21:226, 21:378,<br>21:419, 21:457, 21:468, 21:579,<br>21:611, 21:638, 21:646, 21:652,<br>21:656, 21:662, 21:666, 21:672,<br>21:678, 21:685, 21:686, 21:692,<br>21:696, 21:746, 21:792, 21:1244,<br>21:1262, 21:1268, 21:1272, 21:1278,<br>21:1282, 21:1288, 21:1294, 21:1301,<br>21:1302, 21:1308, 21:1312, 21:1314,<br>21:1421, 24:441, 33:171, 33:198, 57:519                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \#                | 01:46, 01:59, 02:6, 02:14,<br>02:531, 06:801, 09:446, 24:428, 57:502                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| \\$               | 01:58, 02:4, 02:13, 06:800,<br>21:307, 21:444, 21:451, 21:565,<br>21:804, 21:811, 33:631, 33:638, 57:503                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \\$               | <u>1212</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \%                | 01:59, 01:89,<br>01:91, 01:111, 02:14, 02:529, 06:801,<br>21:491, 21:493, 24:430, 33:640,<br>33:760, 37:210, 50:1566, 50:1567, 57:504                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \&                | 01:58, 02:5,<br>02:13, 02:530, 06:800, 50:439, 57:505                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \'                | 02:551, 21:227,<br>21:379, 21:421, 21:455, 21:465,<br>21:581, 21:591, 21:597, 21:599,<br>21:602, 21:604, 21:612, 21:618,<br>21:624, 21:626, 21:629, 21:631,<br>21:639, 21:643, 21:650, 21:654,<br>21:659, 21:664, 21:667, 21:669,<br>21:676, 21:681, 21:682, 21:689,<br>21:694, 21:697, 21:747, 21:794,<br>21:813, 21:815, 21:816, 21:817,<br>21:820, 21:822, 21:823, 21:824,<br>21:826, 21:827, 21:1238, 21:1259,<br>21:1266, 21:1270, 21:1275, 21:1280,<br>21:1283, 21:1285, 21:1292, 21:1297,<br>21:1298, 21:1305, 21:1310, 21:1313,<br>21:1321, 21:1322, 21:1369, 21:1370,<br>21:1375, 21:1376, 21:1387, 21:1388,<br>21:1393, 21:1394, 21:1422, 21:1423,<br>21:1437, 21:1438, 21:1439, 21:1440,<br>21:1453, 21:1454, 24:440, 29:775,<br>30:260, 33:161, 37:738, 38:254,<br>40:387, 40:408, 41:72, 54:588, 57:520 |
| \`                | 07:2364, 38:345                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \`                | 02:551, 07:2386, 38:346                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| \`                | <u>38:271</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \*                | 24:433, 50:1247, 50:1378, 50:1492, 50:1568                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| \*                | <u>38:251</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \+                | 41:72                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \+                | <u>1210</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \,                | 02:432, 02:434, 30:513, 37:738,<br>38:7, 38:8, 38:40, 38:167, 38:169,<br>38:172, 38:197, 38:220, 38:221, 38:237                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| \,                | <u>1251</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \-                | 02:378, 06:24, 18:463,<br>18:554, 21:416, 21:417, 21:574,<br>21:788, 21:789, 24:435, 37:738,<br>40:386, 40:407, 41:72, 57:224, 57:264                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| \-                | <u>06:835, 110</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| \.                | 02:431, 02:433, 20:45, 20:115,<br>20:173, 21:228, 21:380, 21:452,<br>21:453, 21:474, 21:587, 21:588,<br>21:614, 21:615, 21:641, 21:748,<br>21:818, 21:825, 21:1243, 21:1325,<br>21:1326, 21:1335, 21:1336, 21:1345,<br>21:1346, 21:1363, 21:1424, 21:1425,<br>21:1461, 21:1462, 24:434, 33:173, 33:199                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| \.                | <u>1239</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \.default         | <u>552</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| .code             | <u>51:4</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| .if               | <u>51:4</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| .ifnot            | <u>51:4</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| .pass-to-packages | <u>51:22</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| .store            | <u>51:4</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| .usage            | <u>51:4</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \`                | 01:81, 06:25,<br>09:202, 09:277, 24:382, 24:436, 50:438                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| \`                | <u>1206</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| \:                | <u>38:214, 38:224,</u><br><u>38:225, 38:226, 38:242, 38:252,</u><br><u>38:252, 06:795, 06:796, 02:432, 02:434</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| \;                | 02:432, 02:434, 30:507, 38:198                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| \;                | <u>38:214, 791</u>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| \<                | 21:575, 21:739, 24:431, 37:738, 41:71, 41:109                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| \=                | 21:229, 21:381, 21:473, 21:749,<br>21:1241, 21:1315, 21:1316, 21:1331,<br>21:1332, 21:1354, 21:1355, 21:1356,<br>21:1381, 21:1382, 21:1407, 21:1408,<br>21:1441, 21:1442, 21:1443, 21:1444,<br>21:1459, 21:1460, 21:1467, 21:1468,<br>29:775, 33:179, 40:387, 40:408, 41:71                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltypen.dtx, 57=ltfinal.dtx

```

\> 21:572, 21:740, 24:432, 37:738, \@ifpackagelater 977
 38:226, 38:241, 38:242, 38:252, 41:71 \@ifpackageloaded 977
\? 02:431, 02:433, 57:520 \ifpackagewith 977
@C commands: \ifstar 81
 \g_@@_{hook}_code_prop 1270 \ifundefined 81
 \@@_cmd_type_cases:NnnnnTF 1261 \missingfileerror 978
 \c_@@_hash_tl 1264 \namedef 81
 \c_@@_hashes_tl 1264 \nameuse 81
 \@@_if_file_hook:wTF 1260 \restorepar 399
 \@@_include_legacy_code_chunk:n 1266 \setpar 399
 \g_@@_input_file_seq 1257 @topnum commands:
 \@@_print_args:nn 1267 \topnum: 54:1053
 \@@_replacing_args_false: 1266 \l[..... 24:437, 29:43,
 \@@_set_curr_file:nNN 52:386 29:54, 29:76, 29:87, 38:347, 57:521
 \@@_set_curr_file_assign:nnNN 52:386 \l[..... 38:289, 38:462, 1241
 \g_@@_standard_everypar_tl 1269 \\\ 01:58, 01:231, 01:232, 01:233,
 \@@_start_expandable:nNNNNn 1264 01:234, 01:237, 01:244, 01:245,
 \@@textvisiblespace_\meta_{hook} 225 01:246, 01:247, 01:250, 01:257,
 \@@next\textvisiblespace_\meta_{hook} 225 01:258, 01:259, 01:260, 01:263,
 \@@toplevel\textvisiblespace_\meta_{hook} 225 01:270, 01:276, 01:277, 01:281,
 \@@par 399 01:283, 01:284, 01:288, 01:293,
 \@TeXversion 6, 1 01:294, 01:297, 01:303, 02:13,
 @botlist commands: 02:377, 04:272, 04:424, 06:231,
 \@botlist: 54:1033, 54:1127, 54:1286 06:288, 06:449, 06:572, 06:590,
 @botnum commands: 06:800, 07:468, 07:532, 07:563,
 \@botnum: 54:1007 07:704, 07:716, 07:747, 07:1118,
 \@car 82 07:1124, 07:2547, 07:2575, 07:2589,
 \@cdr 82 07:2596, 07:2625, 07:2722, 07:2750,
 @citeb commands: 07:2761, 07:2787, 07:2793, 07:2800,
 \@citeb: 47:36, 47:65, 47:82 08:1777, 08:1788, 08:2580, 08:2594,
 @colht commands: 08:2595, 08:2596, 08:2597, 08:2598,
 \@colht: 54:545, 54:559 08:2603, 08:2606, 08:2607, 08:2608,
 @colnum commands: 08:2618, 08:2619, 08:2633, 08:2654,
 \@colnum: 54:1192, 54:1360 08:2658, 08:2659, 08:2661, 08:2667,
 \@cons 82 08:2678, 08:2679, 08:2684, 08:2689,
 \@currdir 5, 1 08:2694, 08:2710, 08:2875, 08:2876,
 @currname commands: 08:2877, 09:443, 09:581, 11:965,
 \@currname: 20:761 11:973, 11:979, 11:980, 11:981,
 \@dblarg 81 11:982, 11:995, 11:1001, 11:1002,
 @dbldeflist commands: 11:1008, 11:1014, 11:1015, 11:1021,
 \@dbldeflist: 54:1759, 54:1801 11:1022, 11:1029, 11:1036, 11:1043,
 @dbltopnum commands: 11:1050, 11:1051, 11:1060, 11:1078,
 \@dbltopnum: 54:1658, 54:1785 11:1079, 11:1109, 11:1110, 11:1111,
 @deferlist commands: 11:1112, 11:1113, 11:1114, 11:1115,
 \@deferlist: 54:1118, 54:1215, 11:1116, 11:1117, 11:1118, 11:1119,
 54:1276, 54:1384, 54:1428, 54:1457, 14:279, 16:146, 16:153, 16:160,
 54:1515, 54:1547, 54:1633, 54:1673 16:161, 16:162, 16:163, 18:556,
 \@ifclasslater 977 20:754, 20:773, 21:560, 24:425,
 \@ifclassloaded 977 30:258, 36:248, 37:451, 37:456,
 \@ifclasswith 977 37:461, 37:471, 37:475, 37:479,
 \@ifdefinable 81 37:520, 38:370, 38:523, 40:419,
 \@ifnextchar 81 40:578, 40:580, 41:73, 41:170,
 41:180, 41:194, 42:139, 57:506, 421
 \\\ 18:53, 1223

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\{ 01:6, 01:10, 01:58,
 02:2, 02:13, 07:635, 07:1918, 09:444,
 14:22, 21:308, 21:562, 24:426,
 30:256, 37:519, 38:59, 38:167, 57:509
\} 01:11, 01:58, 02:3,
 02:13, 09:445, 14:21, 21:309, 21:563,
 24:427, 30:257, 37:519, 38:59, 57:510
\langle addto-cmd\rangle 196
\langle cmd\rangle_ 146
\langle cmd\rangle_{arg_(num)} 148
\langle cmd\rangle_{uu} 146
\langle cmd\rangle_code 146
\langle cmd\rangle_defaults 146
\langle filename\rangle 1051
\langle function\rangle 168
\lrcorner 01:58,
 01:75, 02:13, 02:381, 02:437, 02:466,
 02:467, 02:484, 06:800, 08:2659,
 11:1002, 11:1015, 11:1022, 11:1030,
 11:1051, 11:1079, 14:19, 14:20,
 14:21, 14:22, 14:25, 18:468, 24:422,
 24:672, 24:709, 24:734, 30:259,
 37:503, 37:504, 37:513, 37:514,
 37:659, 37:675, 37:689, 43:53, 43:55,
 43:60, 43:62, 47:37, 50:432, 57:501
\] 02:551, 24:438, 38:348, 57:522
\] 38:289, 38:486
\` .. 01:47, 01:56, 01:59, 01:103, 01:314,
 02:7, 02:9, 02:11, 02:14, 02:380,
 02:437, 02:438, 02:457, 02:458,
 02:479, 02:480, 06:20, 06:801, 07:6,
 07:7, 07:1857, 07:2232, 07:2237,
 07:3015, 18:556, 18:558, 18:560,
 21:231, 21:286, 21:382, 21:456,
 21:466, 21:558, 21:644, 21:651,
 21:655, 21:660, 21:665, 21:670,
 21:677, 21:683, 21:684, 21:690,
 21:695, 21:750, 21:1239, 21:1256,
 21:1260, 21:1267, 21:1271, 21:1276,
 21:1281, 21:1286, 21:1293, 21:1299,
 21:1300, 21:1306, 21:1311, 21:1323,
 21:1324, 21:1341, 21:1342, 21:1349,
 21:1350, 21:1364, 21:1365, 21:1366,
 21:1395, 21:1396, 21:1417, 21:1418,
 21:1419, 21:1420, 24:423, 24:424,
 24:429, 33:169, 33:197, 37:202,
 37:211, 37:505, 37:506, 37:660,
 50:1248, 50:1249, 50:1250, 50:1331,
 50:1334, 50:1337, 50:1379, 50:1380,
 50:1381, 50:1463, 50:1466, 50:1469,
 50:1493, 50:1494, 50:1495, 50:1553,
 50:1556, 50:1559, 57:255, 57:256,
 57:257, 57:258, 57:259, 57:260,
 57:261, 57:262, 57:263, 57:507,
 57:513, 57:514, 57:515, 57:516,
 57:550, 57:551, 57:552, 57:553,
 57:554, 57:555, 57:556, 57:557, 57:558
__ 21:314, 30:261, 38:269, 38:270,
 02:8, 02:14, 06:801, 57:508, 01:59, 1219
\` 21:232, 21:383,
 21:420, 21:454, 21:464, 21:580,
 21:642, 21:649, 21:653, 21:658,
 21:663, 21:668, 21:675, 21:679,
 21:680, 21:688, 21:693, 21:751,
 21:793, 21:1237, 21:1258, 21:1265,
 21:1269, 21:1274, 21:1279, 21:1284,
 21:1291, 21:1295, 21:1296, 21:1304,
 21:1309, 24:439, 29:775, 33:175,
 37:738, 40:387, 40:408, 41:72, 57:523
\| 21:561,
 22:217, 22:228, 30:580, 30:581, 57:524
\~ 01:59,
 02:10, 02:14, 06:801, 14:20, 18:469,
 21:239, 21:287, 21:384, 21:467,
 21:559, 21:645, 21:657, 21:661,
 21:671, 21:687, 21:691, 21:752,
 21:1240, 21:1257, 21:1261, 21:1273,
 21:1277, 21:1287, 21:1303, 21:1307,
 21:1351, 21:1352, 21:1353, 21:1405,
 21:1406, 33:187, 33:207, 37:653,
 37:670, 37:684, 37:699, 37:742, 57:511

```

**A**

```

\A 57:252, 57:526, 57:547
\`a 21:223, 41:1, 57:243, 57:527, 57:538, 1222
\AA 21:240, 21:428, 21:526, 02:443, 1234
\aa 21:245, 21:422, 21:536, 02:443, 1234
\abovedisplayshortskip 38:531, 02:413
\abovedisplayskip 38:524, 38:526, 38:528,
 38:529, 38:530, 38:531, 02:412, 1220
\abspage 750
abspage 36:232, 750
\accent 21:73,
 21:393, 21:423, 21:479, 21:763, 1250
\ActivateGenericHook 08:2734, 08:2736, 08:2741, 08:2743, 213
\active 01:103, 37:503,
 37:504, 37:505, 37:506, 37:513,
 37:514, 37:652, 37:659, 37:660,
 37:669, 37:675, 37:683, 37:689,
 37:698, 37:740, 38:254, 38:269,
 50:1248, 50:1249, 50:1250, 50:1331,
 50:1334, 50:1337, 50:1379, 50:1380,
 50:1381, 50:1463, 50:1466, 50:1469,
 50:1493, 50:1494, 50:1495, 50:1553,
 50:1556, 50:1559, 57:255, 57:256,
 57:257, 57:258, 57:259, 57:260,
 57:261, 57:262, 57:263, 57:507,
 57:513, 57:514, 57:515, 57:516,
 57:550, 57:551, 57:552, 57:553,
 57:554, 57:555, 57:556, 57:557, 57:558

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

02:11, 54:588, 01:48, 02:457, 02:458,  
 02:466, 02:467, 02:479, 02:480, 02:484  
`\acute` ..... 30:523  
 add commands:  
     `add_to_callback` ..... 04:817  
`\add_to_callback` ..... 45  
`\addcontentsline` .....  
     ... 44:70, 44:80, 44:159, 45:16, 1219  
`\AddEverypageHook` ..... 1066  
`\Adding` ..... 1195  
`\addpenalty` .....  
     ... 18:285, 39:124, 39:194, 39:199,  
         44:50, 54:340, 54:1157, 54:1323, 433  
`\AddThispageHook` ..... 1066  
`\addtocontents` .....  
     ... 44:164, 44:171, 44:177, 44:180, 1219  
`\addtocounter` ..... 512  
`\addtocounter` ..... 22:6, 22:18, 1207  
`\AddToHook` ..... 08:2746, 08:2896,  
     26:152, 37:42, 37:43, 37:44, 37:107,  
     37:108, 37:109, 37:396, 37:397,  
     37:398, 37:399, 47:72, 50:1128,  
     50:1129, 50:1130, 52:561, 52:563,  
     52:573, 52:574, 52:575, 52:576,  
     53:190, 53:473, 53:496, 55:7, 305  
`\AddToHookNext` .....  
     ... 08:2757, 08:2894, 52:562, 53:497, 199  
`\AddToHookNextWithArguments` 08:2757, 199  
`\AddToHookWithArguments` ... 08:2746, 197  
`\addtolength` ..... 521  
`\addtolength` ..... 23:16, 38:526, 38:528  
`\AddToNoCaseChangeList` ..... 57:571, 1262  
`\addtoversion` ..... 27:20, 27:139  
`\addvspace` 18:234, 37:429, 39:124, 39:195,  
     39:196, 39:200, 39:248, 44:50, 1206  
`\adjdemerits` ..... 02:336  
`\AE` .... 21:241, 21:398, 21:527, 21:768,  
     21:1127, 21:1437, 21:1441, 57:649  
`\ae` .... 21:246, 21:401, 21:537, 21:772,  
     21:1133, 21:1439, 21:1443, 57:649  
`\afterassignment` ..... 21:212,  
     21:220, 24:328, 38:199, 06:268,  
     06:274, 06:317, 02:501, 02:504, 93  
`\AfterEndEnvironment` ..... 37:392, 218  
`\aftergroup` .... 24:91, 24:347, 24:358,  
     24:362, 26:202, 26:268, 28:114,  
     28:121, 28:129, 28:404, 32:64,  
     37:656, 37:673, 37:687, 37:702,  
     38:445, 38:446, 39:149, 40:187,  
     54:608, 54:609, 54:666, 54:667, 811  
`\AfterLastShipout` ..... 52:573  
`\afterpreamble` ..... 446  
`\aleph` ..... 30:315

**File Key:** 01=`ltdirchk.dtx`, 02=`ltpplain.dtx`, 03=`ltvers.dtx`, 04=`ltluatex.dtx`, 05=`ltexpl.dtx`,  
 06=`ltdefns.dtx`, 07=`ltcmd.dtx`, 08=`lthooks.dtx`, 09=`ltcmhooks.dtx`, 10=`ltsockets.dtx`,  
 11=`lttemplates.dtx`, 12=`ltalloc.dtx`, 13=`ltcntrl.dtx`, 14=`lterror.dtx`, 15=`ltpar.dtx`,  
 16=`ltpara.dtx`, 17=`ltmeta.dtx`, 18=`ltspaced.dtx`, 19=`ltlogos.dtx`, 20=`ltfiles.dtx`,  
 21=`ltoutenc.dtx`, 22=`ltcounts.dtx`, 23=`ltlength.dtx`, 24=`ltfssbas.dtx`, 25=`ltfssaxes.dtx`,  
 26=`ltfsstrc.dtx`, 27=`ltfscmp.dtx`, 28=`ltfssdcl.dtx`, 29=`ltfssini.dtx`, 30=`fontdef.dtx`,  
 31=`preload.dtx`, 32=`ltfntcmd.dtx`, 33=`lttextcomp.dtx`, 34=`ltpageno.dtx`, 35=`ltxref.dtx`,  
 36=`ltproperties.dtx`, 37=`ltmisen.dtx`, 38=`ltmath.dtx`, 39=`ltlists.dtx`, 40=`ltboxes.dtx`,  
 41=`ltab.dtx`, 42=`ltpictur.dtx`, 43=`ltthm.dtx`, 44=`ltsect.dtx`, 45=`ltfloat.dtx`,  
 46=`ltdxglo.dtx`, 47=`ltbibl.dtx`, 48=`ltmarks.dtx`, 49=`ltpage.dtx`, 50=`ltclass.dtx`,  
 51=`ltkeys.dtx`, 52=`ltfilehook.dtx`, 53=`ltshipout.dtx`, 54=`ltoutput.dtx`, 55=`lttagging.dtx`,  
 56=`lthyphen.dtx`, 57=`ltfinal.dtx`

\AtBeginShipoutBox . . . . . 53:494, 1065  
 \AtBeginShipoutDiscard . . . . . 53:500, 1066  
 \AtBeginShipoutFirst 53:456, 53:498, 1066  
 \AtBeginShipoutInit . . . . . 53:495, 1066  
 \AtBeginShipoutNext . 53:454, 53:496, 1066  
 \AtBeginShipoutOriginalShipout . . . . .  
     . . . . . 53:509, 1065  
 \AtBeginShipoutUpperLeft . 53:501, 1065  
 \AtBeginShipoutUpperLeftForeground . . . . .  
     . . . . . 53:501, 1065  
 \AtEndAfterFileList . . . . . 52:575  
 \AtEndDocument 37:144, 50:1119, 52:579, 1238  
 \AtEndDvi . . . . . 53:463, 53:468, 1261  
 \AtEndEnvironment . . . . . 37:392, 218  
 \AtEndOfClass . . . . . 38:461, 50:1119, 1038  
 \AtEndOfPackage . . . . . 50:606, 50:625,  
     50:723, 50:734, 50:1119, 51:85, 1038  
 \AtEndPreamble . . . . . 219  
 \AtNextShipout . . . . . 1066  
 \atopwithdelims . . . . . 38:57, 38:58, 38:59  
 \attribute . . . . . 04:79, 42  
 \attributedef . . . . . 04:79, 04:223  
 \attributezero . . . . . 04:223  
 \AtVeryEndDocument . . . . . 52:574  
 \AtVeryVeryEnd . . . . . 52:576  
 \author . . . . . 898  
 \author . . . . . 44:8, 44:24, 44:32

**B**

\b . . . . . 21:233,  
     21:389, 21:475, 21:759, 21:1248, 1222  
 \backslash . . . . . 30:258, 30:597  
 \bar . . . . . 30:527  
 \baselineskip . . . . . 26:186, 26:187,  
     26:188, 26:190, 26:191, 30:517,  
     38:171, 38:172, 38:191, 38:197,  
     38:201, 40:396, 40:415, 40:601,  
     41:198, 42:136, 42:314, 42:373,  
     53:222, 53:273, 54:244, 54:275,  
     54:626, 54:641, 54:685, 54:700,  
     02:409, 02:436, 02:499, 02:535, 1209  
 \baselinestretch 24:319, 26:120, 26:121,  
     26:166, 26:167, 26:184, 26:245, 1193  
 \baselinestretch . . . . . 1213  
 \batchmode . . . . . 20:691,  
     20:722, 20:723, 27:106, 05:89,  
     05:95, 05:105, 57:667, 57:688, 462  
 \BCPdata . . . . . 57:560, 57:583, 57:587  
 \BeforeBeginEnvironment . . . . . 37:392, 218  
 \BeforeClearDocument . . . . . 52:577  
 \begin . . . . . 09:66, 14:246, 14:248, 20:345,  
     20:397, 26:7, 30:4, 31:4, 37:242,  
     37:243, 37:313, 37:335, 37:360,  
     37:376, 37:384, 38:466, 38:478,

        44:14, 44:17, 50:744, 51:253, 53:394,  
         06:709, 56:3, 07:1426, 07:1536, 770  
 \begingroup . . . . . 444  
 \belowdisplayshortskip . . . . . 38:530, 02:415  
 \belowdisplayskip . . . . . 38:529, 02:414  
 \beta . . . . . 30:276  
 \bezier . . . . . 862  
 \bezier . . . . . 42:681,  
     42:682, 42:804, 42:805, 42:820, 42:822  
 \bf . . . . . 1201  
 \bfdefault 29:15, 29:261, 29:267, 29:268,  
     29:269, 29:270, 29:310, 29:311,  
     29:312, 29:313, 29:322, 29:358,  
     29:382, 29:401, 29:442, 29:476,  
     30:100, 30:111, 30:113, 30:121, 643  
 \bfseries . . . . . 29:13, 29:14, 29:253,  
     29:254, 29:302, 29:307, 29:308,  
     29:349, 29:352, 29:353, 29:378,  
     29:380, 29:381, 29:474, 29:475,  
     32:19, 35:17, 35:34, 35:51, 43:53,  
     43:55, 43:60, 43:62, 47:40, 53:395, 635  
 bfseries . . . . . 29:420  
 bfseries/defaults . . . . . 29:420  
 \bgroup . . . . . 02:450, 1206  
 \bibcite . . . . . 47:7, 47:9, 47:10, 1226  
 \bibdata . . . . . 47:45, 47:49  
 \bibitem . . . . . 47:3  
 \bibliography . . . . . 937  
 \bibliography . . . . . 47:47  
 \bibliographystyle . . . . . 937  
 \bibliographystyle . . . . . 47:52  
 \bibstyle . . . . . 47:45, 47:57  
 \Big . . . . . 30:630, 30:633,  
     30:642, 30:644, 38:44, 38:45, 38:46  
 \big . . . . . 30:631, 30:643, 38:41  
 \bigbreak . . . . . 06:879, 06:900, 02:515  
 \bigcap . . . . . 30:352  
 \bigcirc . . . . . 30:399  
 \bigcup . . . . . 30:353  
 \Bigg . . . . . 30:637, 30:646, 38:50, 38:51, 38:52  
 \bigg . . . . . 30:635, 30:645, 38:47, 38:48, 38:49  
 \Biggl . . . . . 38:50  
 \biggl . . . . . 38:47  
 \Biggm . . . . . 38:51  
 \biggm . . . . . 38:48  
 \Biggr . . . . . 38:52  
 \biggr . . . . . 38:49  
 \Bigl . . . . . 38:44  
 \bigl . . . . . 38:41  
 \Bigm . . . . . 38:45  
 \bigm . . . . . 38:42  
 \bigodot . . . . . 30:360  
 \bigoplus . . . . . 30:359  
 \bigotimes . . . . . 30:358

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\Bigr 38:46
\bigr 38:43
\bigskip 18:447, 02:520
\bigskipamount 18:449, 18:450, 45:391, 02:519
\bigsqcup 30:363
\bigtriangledown 30:368, 30:369
\bigtriangleup 30:367, 30:370
\biguplus 30:351
\bigvee 30:349
\bigwedge 30:350
\binoppenalty 02:323
\bmod 38:35
\boldmath 19:14, 29:613
bool commands:
 \bool_gset_false:N 10:16,
 36:31, 48:313, 53:15, 53:81, 53:90, 08:15
 \bool_gset_true:N ... 10:11, 36:33,
 48:308, 53:10, 53:120, 53:344, 08:10
 \bool_if:NTF
 ... 09:162, 09:166, 09:178, 09:237,
 09:241, 09:253, 09:380, 10:22, 10:24,
 11:222, 11:228, 11:658, 11:761,
 11:810, 11:827, 11:849, 36:71,
 48:319, 51:24, 51:216, 53:21, 53:88,
 53:365, 07:121, 07:130, 07:142,
 07:143, 07:157, 07:169, 07:173,
 07:182, 07:184, 07:194, 07:203,
 07:204, 07:207, 07:212, 07:214,
 07:318, 07:463, 07:472, 07:474,
 07:529, 07:560, 07:575, 07:624,
 07:657, 07:701, 07:711, 07:744,
 07:752, 07:757, 07:758, 07:766,
 07:771, 07:794, 07:807, 07:885,
 07:931, 07:932, 07:991, 07:1004,
 07:1021, 07:1041, 07:1954, 07:2207,
 07:2539, 07:2550, 08:21, 08:340,
 08:1913, 08:1922, 08:2013, 08:2022, 191
 \bool_if_exist:NTF 36:27
 \bool_lazy_and:nnTF 08:2449,
 08:2917, 53:66, 53:112, 53:376,
 07:740, 07:933, 08:275, 08:1915, 08:2015
 \bool_lazy_and_p:nn ... 08:2452, 07:112
 \bool_lazy_any:nTF ... 07:108, 07:483
 \bool_lazy_or:nnTF
 ... 07:73, 07:1385, 07:1394,
 07:2362, 07:2384, 08:886, 08:1866
 \bool_new:N 10:6, 11:21,
 11:22, 36:28, 48:304, 51:57, 53:6,
 53:188, 53:203, 07:16, 07:18, 07:20,
 07:31, 07:32, 07:34, 07:35, 07:37,
 07:40, 07:43, 07:44, 07:45, 08:6, 08:24
 \bool_set_false:N
 ... 11:277, 11:656, 11:737, 51:84,
 55:44, 07:57, 07:83, 07:232, 07:254,
 07:299, 07:449, 07:450, 07:451,
 07:452, 07:453, 07:454, 07:475,
 07:537, 07:567, 07:592, 07:606,
 07:628, 07:629, 07:630, 07:645,
 07:707, 07:750, 07:761, 07:762,
 07:783, 07:784, 07:785, 07:787,
 07:791, 07:798, 07:940, 07:941,
 07:942, 07:1844, 08:1905, 08:2005
 \bool_set_true:N
 ... 11:248, 11:290, 11:615,
 11:746, 51:81, 55:46, 07:62, 07:233,
 07:255, 07:288, 07:448, 07:473,
 07:544, 07:551, 07:552, 07:566,
 07:753, 07:754, 07:814, 07:815,
 07:821, 07:822, 07:828, 07:829,
 07:836, 07:837, 07:838, 07:958,
 07:976, 07:1849, 08:1902, 08:2002
 \bool_while_do:nn ... 08:1592, 08:1677
 \c_false_bool 09:141, 09:151,
 09:370, 07:2208, 07:2511, 07:2802,
 07:2970, 08:397, 08:403, 08:411, 191
 \l_tmpa_bool 346
 \c_true_bool
 ... 09:146, 07:2209, 07:2513, 07:2803,
 07:2968, 08:404, 08:412, 08:414, 313
bool internal commands:
 \g_mark_debug_bool
 ... 48:304, 48:308, 48:313, 48:319
 \BooleanFalse .. 07:1837, 07:2201, 07:2802
 \BooleanTrue ... 07:1836, 07:2200, 07:2802
 \bordermatrix 38:185
 \bot 30:328
 \botfigrule 54:746, 54:2363
 \botmark 49:79, 54:651, 54:710, 942
 \bottomfraction 45:275, 54:2332
 \bowtie 30:489
 \Box 29:726
 \box 404
box commands:
 \box_dp:N 53:194
 \box_gclear:N 16:89
 \box_gset_to_last:N 16:16, 16:49, 16:117
 \box_ht:N 53:193, 53:302, 1070
 \box_if_empty:NTF . 48:86, 53:78, 53:97
 \box_if_horizontal:NTF 53:234, 53:286
 \box_if_vertical:NTF
 ... 48:81, 48:382, 48:413, 53:208, 53:259
 \box_move_up:nn 53:248, 53:302
 \box_new:N 16:84, 48:61,
 48:62, 53:23, 53:25, 53:187, 53:204
 \box_set_dp:Nn ... 53:221, 53:230,
 53:247, 53:272, 53:283, 53:301, 53:332
 \box_set_eq:NN . 53:149, 53:179, 53:184

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\box\_set\_eq\_drop:NN ..... 53:93  
 \box\_set\_ht:Nn ... 53:220, 53:229,  
   53:246, 53:271, 53:282, 53:300, 53:331  
 \box\_set\_to\_last:N ..... 48:78  
 \box\_set\_wd:Nn .....  
   ..... 53:219, 53:245, 53:270, 53:299  
 \box\_use:N ..... 53:126,  
   53:225, 53:250, 53:278, 53:303, 53:333  
 \box\_use\_drop:N ..... 16:86  
 \box\_wd:N ..... 53:195, 53:296, 53:304  
 box internal commands:  
   \l\_mark\_box ... 48:61, 48:74, 48:78,  
   48:81, 48:82, 48:85, 48:86, 48:96, 954  
   \l\_mark\_ii\_box ... 48:61, 48:85, 48:95  
 \boxmaxdepth ... 42:474, 42:502, 42:532,  
   42:610, 42:627, 54:490, 54:510,  
   54:550, 54:719, 54:728, 54:768, 02:395  
 \brace ..... 38:59  
 \braceleft .....  
   30:560, 30:564, 30:565, 30:567, 30:569  
 \bracelu ..... 30:562, 30:566, 30:568  
 \bracerd ..... 30:561, 30:566, 30:568  
 \braceru ..... 30:563, 30:565, 30:569  
 \bracevert ..... 30:615  
 \brack ..... 38:58  
 \break ..... 18:116,  
   06:880, 06:901, 02:508, 02:513, 1227  
 \breve ..... 30:528  
 \brokenpenalty ..... 24:692, 02:328  
 \buildrel ..... 30:476, 38:162  
 \bullet ..... 30:388

**C**

\c ..... 11:383, 21:234, 21:335, 21:337,  
   21:339, 21:341, 21:343, 21:345,  
   21:347, 21:349, 21:351, 21:372,  
   21:374, 21:392, 21:459, 21:478,  
   21:606, 21:608, 21:633, 21:635,  
   21:648, 21:674, 21:701, 21:704,  
   21:705, 21:706, 21:707, 21:708,  
   21:709, 21:710, 21:711, 21:712,  
   21:762, 21:1250, 21:1264, 21:1290,  
   21:1347, 21:1348, 21:1367, 21:1368,  
   21:1371, 21:1372, 21:1377, 21:1378,  
   21:1389, 21:1390, 21:1397, 21:1398,  
   21:1401, 21:1402, 33:167, 33:196, 1243  
 \cal ..... 29:776  
 call commands:  
   call\_callback ..... 04:798  
 \call\_callback ..... 46  
 callback commands:  
   callback\_descriptions ..... 04:983  
 callback.register ..... 04:686  
 \callback\_descriptions ..... 46

\cap ..... 30:379  
 \capitalacute ..... 21:839, 33:160,  
   33:161, 33:193, 33:690, 33:940, 33:1306  
 \capitalbreve ..... 21:846, 33:162,  
   33:163, 33:194, 33:691, 33:947, 33:1313  
 \capitalcaron ..... 21:845, 33:164,  
   33:165, 33:195, 33:692, 33:946, 33:1312  
 \capitalcedilla ..... 21:832, 33:166,  
   33:167, 33:196, 33:693, 33:937, 33:1316  
 \capitalcircumflex ... 21:840, 33:168,  
   33:169, 33:197, 33:694, 33:941, 33:1307  
 \capitaldieresis ... 21:842, 33:170,  
   33:171, 33:198, 33:695, 33:943, 33:1309  
 \capitaldotaccent ... 21:848, 33:172,  
   33:173, 33:199, 33:696, 33:949, 33:1315  
 \capitalgrave ..... 21:838, 33:174,  
   33:175, 33:200, 33:697, 33:939, 33:1305  
 \capitalhungarumlaut . 21:843, 33:176,  
   33:177, 33:201, 33:698, 33:944, 33:1310  
 \capitalmacron ..... 21:847, 33:178,  
   33:179, 33:202, 33:699, 33:948, 33:1314  
 \capitalnewtie .....  
   ... 21:852, 33:190, 33:191, 33:203,  
   33:700, 33:1014, 33:1015, 33:1321, 701  
 \capitalogonek ..... 21:835, 33:180,  
   33:181, 33:204, 33:701, 33:938, 33:1317  
 \capitalring ..... 21:844, 33:182,  
   33:183, 33:205, 33:702, 33:945, 33:1311  
 \capitaltie .....  
   ... 21:850, 33:184, 33:185, 33:206,  
   33:703, 33:1010, 33:1011, 33:1319  
 \capitaltilde ..... 21:841, 33:186,  
   33:187, 33:207, 33:704, 33:942, 33:1308  
 \caption ..... 45:4, 304  
 \cases ..... 38:166, 38:167, 38:177, 38:179  
 \CaseSwitch ..... 57:571, 1262  
 \catcode ..... 767  
 \catcodetable ..... 04:89, 04:109, 42  
 \catcodetable@atletter ..... 43  
 \catcodetable@initex ..... 43  
 \catcodetable@latex ..... 43  
 \catcodetable@string ..... 43  
 \catcoding ..... 1197  
 \cdot ..... 30:401  
 \cdotp ..... 30:509, 30:515  
 \cdots ..... 30:515  
 center (env.) ..... 37:444  
 \center ..... 37:444  
 \centering ..... 37:444, 37:449,  
   37:450, 37:468, 37:470, 37:485, 37:487  
 \centerline ..... 40:614  
 \changes ..... 22:36, 29:667, 76  
 \chaptermark ..... 946

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\char ... 21:391, 21:394, 21:430, 21:433, 30:398
 21:444, 21:451, 21:477, 21:481, 21:486, 21:489, 21:491, 21:493, 21:733, 21:761, 21:764, 21:797, 21:804, 21:811, 21:834, 21:837, 21:866, 21:896, 21:1006, 21:1041, 21:1165, 21:1167, 21:1169, 21:1216, 29:626, 29:633, 33:631, 33:638, 33:640, 33:760, 37:576, 37:743, 38:251, 42:232, 42:282, 42:296, 42:304, 42:307, 42:448, 42:562, 42:567, 42:575, 42:579, 42:615, 42:616, 42:618, 42:631, 42:632, 42:635, 42:662, 06:842, 06:857, 1230
char commands:
 \char_generate:nn 05:145, 07:1666, 07:1962
 \char_set_catcode_active:N . 07:2232
 \char_set_catcode_active:n . 07:2409
 \char_set_catcode_escape:N .. 09:443
 \char_set_catcode_group_begin:N ..
 09:444
 \char_set_catcode_group_end:N 09:445
 \char_set_catcode_other:N .. 07:1952
 \char_set_catcode_other:n .. 07:1955
 \char_set_catcode_parameter:N 09:446
 \char_set_catcode_parameter:n 07:1956
 \char_set_lccode:nn . 07:2237, 07:2419
 \char_value_catcode:n 09:375
\chardef 12:2, 04:22, 04:26, 04:38, 51:65
 04:47, 04:48, 20:52, 20:122, 04:89, 21:18, 04:157, 04:224, 24:14, 41:4, 41:9, 50:1254, 50:1385, 50:1502, 02:10, 02:16, 02:17, 02:18, 02:19, 02:20, 57:42, 57:44, 57:48, 57:67, 57:171, 57:172, 57:173, 57:174, 57:175, 57:176, 57:177, 02:58, 02:64, 02:66, 02:73, 02:79, 02:82, 02:84, 02:94, 02:96, 02:97, 02:98, 02:99, 02:108, 02:114, 02:115, 02:128, 02:130, 02:194, 01:48, 02:253, 02:257, 02:259, 01:54, 02:283, 01:55, 02:298, 02:529, 02:530, 02:531, 1210
\charsubdef 57:353
\charzero 04:224
\check 30:529
\CheckCommand 06:187, 1214
\CheckEncodingSubset 33:16,
 33:61, 33:109, 33:110, 33:111, 33:156, 33:158, 33:352, 33:621, 33:830, 33:879, 33:935, 33:936, 33:1004, 33:1121, 33:1124, 33:1138, 695
\chi 30:295
\choose 38:57
\circ 30:398
\circle 42:450, 42:604, 42:806, 42:823, 1197
\citation 47:11, 47:39, 47:67, 47:84
\cite 937
\cite 47:12, 938
\clap 40:618, 1251
class/.../after 1037
class/.../before 1037
class/after 1037
class/before 1037
\ClassError 14:84
\ClassInfo 14:84
\ClassNote 14:136, 1258
\ClassNoteNoLine 14:136
\ClassWarning 14:84
\ClassWarningNoLine 14:84
\cleaders 30:555, 30:558, 02:549
\cleardoublepage 54:140
\ClearHookNext 08:2768, 199
\ClearHookRule 08:2805, 08:2910, 203
\clearpage 20:297, 20:324,
 20:328, 20:359, 20:382, 20:385, 20:406, 20:424, 37:17, 37:83, 37:147, 37:240, 54:127, 54:140, 54:145, 54:202, 54:409, 54:412, 54:416, 54:457, 54:463, 54:2209, 54:2226, 1039
\cline 41:367, 1269
clist commands:
 \clist_clear:N 51:65
 \clist_gclear:N 08:1591, 08:1676
 \clist_gput_left:Nn . 08:1495, 08:1530
 \clist_gput_right:Nn 08:1497, 08:1534
 \clist_if_empty:NTF 11:534, 11:563, 08:1932, 08:2033
 \clist_if_exist:NTF 08:121
 \clist_if_in:NnTF . 11:266, 11:570, 51:51, 51:52, 51:127, 51:150, 51:152
 \clist_map_function:nN 36:59
 \clist_map_inline:Nn 11:536, 51:108, 51:139, 51:204
 \clist_map_inline:nn 51:239, 08:917, 08:926
 \clist_map_variable:Nnn 51:82
 \clist_new:N 11:32, 51:54, 51:55, 51:56, 08:123, 08:140
 \clist_put_right:Nn . 51:33, 51:53, 51:121, 51:128, 51:135, 51:160, 51:169
 \clist_remove_all:Nn 11:572, 51:35, 51:134, 51:161
 \clist_set:Nn 11:524
 \clist_use:Nn 08:1934, 08:2035
 \l_tmpa_clist 346
\clubpenalty 20:8, 20:25, 20:95, 20:152, 24:690,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

```

39:128, 39:218, 39:220, 44:100,
44:106, 44:130, 44:135, 02:325, 1233
\clubsuit 30:338
cmd internal commands:
 __cmd_add_arg:n 07:1602, 07:1613, 07:1618, 07:1619,
 07:1653, 07:1741, 07:1748, 07:1823,
 07:1836, 07:1837, 07:1911, 07:1968,
 07:1975, 07:2007, 07:2007, 07:2012, 156
 __cmd_add_arg_spec:n ... 07:590,
 07:604, 07:664, 07:738, 07:738, 07:778
 __cmd_add_arg_spec_mandatory:n .
 07:636, 07:646, 07:652, 07:738, 07:764
 __cmd_add_default: 07:846,
 07:884, 07:901, 07:952, 07:955,
 07:962, 07:972, 07:1039, 07:1048, 136
 __cmd_add_default:n
 .. 07:853, 07:893, 07:909,
 07:952, 07:952, 07:969, 07:983, 136
 __cmd_add_default_E:nn
 .. 07:861, 07:952, 07:967, 07:1017
 __cmd_add_expandable_grabber:nn
 07:996, 07:1009, 07:1020,
 07:1040, 07:1050, 07:1057, 07:1057
 __cmd_add_expandable_type_+:w 07:974
 __cmd_add_expandable_type_D:w ..
 07:979, 07:979
 __cmd_add_expandable_type_D_-
 aux:NN 07:979, 07:985, 07:1002
 __cmd_add_expandable_type_D_-
 aux:NNN ... 07:979, 07:986, 07:989
 __cmd_add_expandable_type_D_-
 aux:NNNN
 .. 07:979, 07:980, 07:981, 07:1045
 __cmd_add_expandable_type_E:w ..
 07:1015, 07:1015
 __cmd_add_expandable_type_E_-
 aux:n ... 07:1015, 07:1019, 07:1031
 __cmd_add_expandable_type_m:w ..
 07:1037, 07:1037
 __cmd_add_expandable_type_R:w ..
 07:1044, 07:1044
 __cmd_add_expandable_type_t:w ..
 07:1046, 07:1046
 __cmd_add_grabber:N 07:847,
 07:854, 07:867, 07:886, 07:894,
 07:902, 07:910, 07:924, 07:924, 138
 __cmd_add_type_!:w 07:818
 __cmd_add_type_+:w 07:811
 __cmd_add_type_=: 07:833
 __cmd_add_type_>:w 07:825
 __cmd_add_type_b:w .. 07:843, 07:843
 __cmd_add_type_D:w .. 07:850, 07:850
 __cmd_add_type_E:w .. 07:858, 07:858
 __cmd_add_type_m:w .. 07:882, 07:882
 __cmd_add_type_R:w .. 07:890, 07:890
 __cmd_add_type_t:w .. 07:898, 07:898
 __cmd_add_type_v:w .. 07:906, 07:906
 __cmd_all_m_check:n
 07:96, 07:96, 07:116
 __cmd_all_m_check_aux:n
 07:96, 07:97, 07:98
 __cmd_allowed_token_check:N ...
 07:588,
 07:600, 07:621, 07:642, 07:680, 07:680
 \l__cmd_arg_spec_tl
 07:11, 07:90, 07:447,
 07:535, 07:568, 07:622, 07:755, 113
 __cmd_arg_to_keyvalue:nn
 07:840, 07:2291, 07:2291
 __cmd_arg_to_keyvalue_auxi:nnn .
 07:2291, 07:2306, 07:2308
 __cmd_arg_to_keyvalue_auxii:Nnnn
 07:2291, 07:2311, 07:2314
 __cmd_arg_to_keyvalue_auxiii:nnn
 07:2291, 07:2317, 07:2320
 __cmd_arg_to_keyvalue_auxiv:Nnnn
 07:2291, 07:2323, 07:2326
 __cmd_arg_to_keyvalue_auxv:nn ..
 07:2291, 07:2312,
 07:2318, 07:2324, 07:2330, 07:2332
 __cmd_arg_to_keyvalue_braces:nnn
 07:2291, 07:2293, 07:2296
 __cmd_arg_to_keyvalue_loop:w ...
 .. 07:2291, 07:2334, 07:2337,
 07:2349, 07:2351, 07:2366, 07:2387
 __cmd_arg_to_keyvalue_loop_-
 group:n .. 07:2291, 07:2343, 07:2348
 __cmd_arg_to_keyvalue_loop_N_-
 type:N .. 07:2291, 07:2340, 07:2352
 __cmd_arg_to_keyvalue_loop_-
 space:w .. 07:2291, 07:2344, 07:2350
 __cmd_arg_to_keyvalue_math:w ...
 07:2291, 07:2365,
 07:2369, 07:2388, 07:2391, 07:2393
 __cmd_arg_to_keyvalue_math_-
 group:n .. 07:2291, 07:2375, 07:2390
 __cmd_arg_to_keyvalue_math_N_-
 type:N .. 07:2291, 07:2372, 07:2380
 __cmd_arg_to_keyvalue_math_-
 space:w .. 07:2291, 07:2376, 07:2392
 __cmd_arg_to_keyvalue_set_-
 default:nn
 .. 07:2291, 07:2355, 07:2383, 07:2394, 177
 __cmd_arg_to_keyvalue_set_-
 keyvalue:nn
 .. 07:2291, 07:2359, 07:2396, 177

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\l__cmd_args_i_t1 .. 07:13, 07:327,
 07:333, 07:338, 07:339, 07:341, 113
\l__cmd_args_ii_t1 .. .
 . 07:14, 07:337, 07:339,
 07:341, 07:379, 07:384, 07:391, 113
\l__cmd_args_process: .. .
 . 07:317, 07:377, 07:377, 137
\l__cmd_args_process_aux:n .. .
 . 07:377, 07:390, 07:394
\l__cmd_args_process_loop:nn .. .
 . 07:377, 07:383, 07:386
\l__cmd_args_t1 .. 07:12, 07:280,
 07:305, 07:322, 07:327, 07:333,
 07:352, 07:381, 07:384, 07:397,
 07:398, 07:1538, 07:1539, 07:1544,
 07:1547, 07:1719, 07:1753, 07:2009, 125
\l__cmd_bad_arg_spec:wn .. 07:505,
 07:510, 07:519, 07:528, 07:574,
 07:587, 07:597, 07:598, 07:603,
 07:614, 07:620, 07:641, 07:732, 07:732
\l__cmd_bad_def:wn .. .
 . 07:461, 07:469, 07:498, 07:533,
 07:564, 07:579, 07:661, 07:669,
 07:677, 07:696, 07:705, 07:717,
 07:732, 07:737, 07:748, 07:775, 145
\l__cmd_bool_reverse:N .. .
 . 07:2205, 07:2205, 07:3001
\l__cmd_break_point:n .. .
 . 07:93, 07:95, 07:95, 07:732,
 07:737, 07:1107, 07:1114, 07:1325
\l__cmd_cant_copy:nwn .. .
 . 07:1104, 07:1114,
 07:1114, 07:1230, 07:1293, 07:1323
\l__cmd_check_definable:nTF .. .
 . 07:2406, 07:2406,
 07:2806, 07:2819, 07:2832, 07:2837,
 07:2925, 07:2938, 07:2951, 07:2959
\l__cmd_check_definable_aux:nN .. .
 . 07:2406, 07:2407, 07:2410, 180
\l__cmd_check_end:n .. .
 . 07:1289, 07:1291, 07:1297
\l__cmd_check_end:Nn .. 07:1276,
 07:1277, 07:1289, 07:1289, 07:1435
\l__cmd_check_end:w .. .
 . 07:1289, 07:1299, 07:1302
\l__cmd_chk_if_free_cs:N .. .
 . 07:3006, 07:3007
\l__cmd_cmd_if_xparse:NTF .. 1257
\l__cmd_cmd_if_xparse_aux:N .. 07:2473
\l__cmd_cmd_type_cases:NnnnnnTF ..
 . 07:1098,
 07:1317, 07:2473, 07:2473, 07:2496, 145
\l__cmd_copy:NN .. .
 . 07:1091, 07:1093, 07:1093,
 . 07:1128, 07:1143, 07:1189, 07:1307, 145
\l__cmd_copy_command:nnNN .. .
 . 07:1099, 07:1129, 07:1129, 146
\l__cmd_copy_command:NnNnNnnn .. .
 . 07:1129, 07:1134, 07:1136, 146
\l__cmd_copy_environment:nnNN .. .
 . 07:1102, 07:1263, 07:1263
\l__cmd_copy_environment:Nnnnnnn .. .
 . 07:1263, 07:1269, 07:1272
\l__cmd_copy_environment_end:nnNN .. .
 . 07:1103, 07:1274, 07:1274
\l__cmd_copy_environment_end_-aux:nnNN .. 07:1274, 07:1278, 07:1281
\l__cmd_copy_expandable:nnN .. .
 . 07:1208, 07:1213, 07:1216,
 07:1243, 07:1253, 07:1259, 07:1261
\l__cmd_copy_expandable:nnNN .. .
 . 07:1100, 07:1142, 07:1146,
 07:1148, 07:1161, 07:1163, 07:1174
\l__cmd_copy_expandable:NnNNNNnnm .. .
 . 07:1142, 07:1158, 07:1171, 07:1191, 146
\l__cmd_copy_expandable_signature:NnNNNNnnn .. .
 . 07:1154, 07:1169, 07:1208, 07:1208, 146
\l__cmd_copy_grabber_{type}:w .. 148
\l__cmd_copy_grabber_D:w .. .
 . 07:1233, 07:1233, 07:1246, 07:1247
\l__cmd_copy_grabber_D_alt:w .. .
 . 07:1233, 07:1245, 07:1248
\l__cmd_copy_grabber_E:w .. .
 . 07:1233, 07:1249, 07:1255
\l__cmd_copy_grabber_E_long:w .. .
 . 07:1233, 07:1255
\l__cmd_copy_grabber_m:w .. .
 . 07:1233, 07:1261, 07:1262
\l__cmd_copy_grabber_m_long:w .. .
 . 07:1233, 07:1262
\l__cmd_copy_grabber_R:w .. .
 . 07:1233, 07:1247
\l__cmd_copy_grabber_R_alt:w .. .
 . 07:1233, 07:1248
\l__cmd_copy_grabber_t:w .. .
 . 07:1233, 07:1256
\l__cmd_copy_optimized:nnNN .. .
 . 07:1101, 07:1177, 07:1177
\l__cmd_copy_parse_grabber:w .. .
 . 07:1208, 07:1220, 07:1224, 148
\l__cmd_current_arg_int .. 07:15,
 07:25, 07:146, 07:154, 07:183,
 07:190, 07:270, 07:351, 07:355,
 07:360, 07:361, 07:445, 07:457,
 07:480, 07:536, 07:582, 07:607,
 07:782, 07:957, 07:964, 07:1211,
 07:1219, 07:1237, 07:1241, 07:1242,
 07:1252, 07:1444, 07:1510, 07:1519, 149

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__cmd_declare_cmd:Nnn
 07:55, 07:55,
 07:2814, 07:2822, 07:2833, 07:2838
__cmd_declare_cmd_aux:Nnn . . .
 07:55, 07:58, 07:63, 07:65
__cmd_declare_cmd_code:Nnn . . .
 07:91, 07:106, 07:106
__cmd_declare_cmd_code_aux:Nnn . .
 07:106, 07:123, 07:150
__cmd_declare_cmd_code_expandable:Nnn
 07:106, 07:122, 07:178
__cmd_declare_cmd_internal:Nnnn
 07:55, 07:84, 07:86, 07:263, 122
__cmd_declare_cmd_optimized:Nnn
 07:106, 07:125, 07:128
__cmd_declare_env:nmmn
 07:224, 07:224,
 07:226, 07:236, 07:238, 07:240,
 07:2856, 07:2871, 07:2877, 07:2887,
 07:2902, 07:2908, 07:2912, 07:2914, 189
__cmd_declare_env_internal:nnnn
 07:224, 07:234, 07:256, 07:261
__cmd_declare_expandable_-
 cmd:Nnn
 07:55, 07:60,
 07:2933, 07:2941, 07:2954, 07:2960
__cmd_defaults: 07:316, 07:324, 07:324
__cmd_defaults_aux:
 07:324, 07:328, 07:329, 07:330, 07:335
__cmd_defaults_bool
 07:16, 07:169,
 07:184, 07:207, 07:787, 07:958, 114
__cmd_defaults_def:
 07:324, 07:326, 07:348
__cmd_defaults_def:nn
 07:324, 07:353, 07:358
__cmd_defaults_def:nnn
 07:324, 07:361, 07:363
__cmd_defaults_error:w
 07:324, 07:331, 07:343
__cmd_defaults_tl
 07:17,
 07:170, 07:191, 07:308, 07:316,
 07:352, 07:788, 07:959, 07:965, 114
__cmd_delimiter_check:nnn . . .
 07:635, 07:644, 07:721, 07:721
__cmd_end_expandable:NNw . . .
 07:404, 07:406, 07:406
__cmd_end_expandable_aux:nNNNN . .
 07:406, 07:409, 07:410
__cmd_end_expandable_aux:w . . .
 07:406, 07:407, 07:408
__cmd_end_expandable_defaults:nmnNn
 07:406, 07:413,
 07:422, 07:431, 07:441, 07:442, 126
__cmd_end_expandable_defaults:nnw
 07:406, 07:430, 07:434
__cmd_end_expandable_defaults:nw
 07:406, 07:437, 07:438, 07:440
__cmd_environment_bool
 07:18, 07:83, 07:110, 07:157,
 07:233, 07:255, 07:288, 07:299,
 07:318, 07:474, 07:657, 07:2550, 114
__cmd_environment_or_command: . .
 ... 07:346, 07:460, 07:492, 07:496,
 07:578, 07:660, 07:667, 07:676,
 07:691, 07:735, 07:769, 07:1821,
 07:1965, 07:1972, 07:2548, 07:2548
__cmd_environment_str
 07:19, 07:160, 07:228,
 07:242, 07:243, 07:244, 07:245,
 07:248, 07:252, 07:257, 07:287,
 07:290, 07:291, 07:319, 07:2551, 114
__cmd_expandable_aux_name_tl . .
 07:21, 07:22,
 07:994, 07:998, 07:1007, 07:1011, 114
__cmd_expandable_bool
 07:20, 07:57, 07:62, 07:130, 07:142,
 07:182, 07:194, 07:232, 07:254,
 07:463, 07:472, 07:529, 07:560,
 07:701, 07:711, 07:744, 07:794, 114
__cmd_expandable_grab_D:nnNNNwNN
 07:2013, 07:2037, 07:2043
__cmd_expandable_grab_D:NNNwNNn
 07:2013, 07:2014, 07:2017
__cmd_expandable_grab_D:NNNwNNnn
 07:2013,
 07:2028, 07:2035, 07:2061, 07:2156, 170
__cmd_expandable_grab_D:Nw . .
 07:2013, 07:2039, 07:2042
__cmd_expandable_grab_D:w . .
 07:2013, 07:2013
__cmd_expandable_grab_D_-
 alt:NNwn . 07:2080, 07:2087, 07:2183
__cmd_expandable_grab_D_-
 alt:NNwNNn 07:2065, 07:2066, 07:2069
__cmd_expandable_grab_D_alt:Nwn
 07:2065
__cmd_expandable_grab_D_alt:w ..
 07:2065, 07:2065
__cmd_expandable_grab_E:w . .
 07:2100, 07:2100
__cmd_expandable_grab_E_aux:w ..
 07:2100,
 07:2101, 07:2103, 07:2104, 07:2132
__cmd_expandable_grab_E_end:nnw
 07:2100, 07:2116, 07:2133
__cmd_expandable_grab_E_-
 find:nnw . 07:2100, 07:2130, 07:2131

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```
__cmd_expandable_grab_E_find:w .
..... 07:2100, 07:2121, 07:2129
_cmd_expandable_grab_E_long:w .
..... 07:2100, 07:2102
_cmd_expandable_grab_E_-
loop:nnnNNw
.. 07:2100, 07:2108, 07:2112, 07:2123
_cmd_expandable_grab_E_-
test:nnw . 07:2100, 07:2105, 07:2106
_cmd_expandable_grab_m:w
..... 07:2135, 07:2135, 149
_cmd_expandable_grab_m_aux:wNn
.. 07:2135, 07:2136, 07:2138, 07:2139
_cmd_expandable_grab_m_long:w .
..... 07:2135, 07:2137
_cmd_expandable_grab_R:w
..... 07:2141, 07:2141
_cmd_expandable_grab_R_alt:w ..
..... 07:2168, 07:2168
_cmd_expandable_grab_R_alt_-
aux:NNwNNn 07:2168, 07:2169, 07:2172
_cmd_expandable_grab_R_-
aux:NNNwNNn 07:2141, 07:2142, 07:2145
_cmd_expandable_grab_t:w
..... 07:2195, 07:2195
_cmd_expandable_grab_t_-
aux:NNwn . 07:2195, 07:2196, 07:2197
_cmd_flush_m_args: 07:794,
07:813, 07:820, 07:827, 07:835,
07:845, 07:852, 07:860, 07:892,
07:900, 07:908, 07:913, 07:913, 139
_cmd_fn_code_tl
..... 07:29, 07:307, 07:322, 114
_cmd_fn_tl
... 07:28, 07:119, 07:306, 07:1540,
07:1607, 07:1631, 07:1637, 07:1647,
07:1657, 07:1666, 07:1671, 07:1675,
07:1706, 07:1732, 07:1740, 07:1742,
07:1747, 07:1749, 07:1760, 07:1761,
07:1766, 07:1767, 07:1772, 07:1773,
07:1778, 07:1779, 07:1784, 07:1785,
07:1790, 07:1791, 07:1796, 07:1797,
07:1802, 07:1803, 07:1833, 07:1839, 156
_cmd_function_tl
.... 07:24, 07:30, 07:88, 07:120,
07:136, 07:139, 07:153, 07:164,
07:165, 07:181, 07:189, 07:199,
07:202, 07:206, 07:208, 07:216,
07:222, 07:468, 07:532, 07:563,
07:704, 07:716, 07:747, 07:2554, 115
_cmd_get_grabber:NN
07:1033, 07:1049, 07:1062, 07:1062, 142
_cmd_get_grabber_auxi:NN
.. 07:1062, 07:1065, 07:1068, 07:1076
_cmd_get_grabber_auxii:NN
..... 07:1062, 07:1083, 07:1086
_cmd_grab_b:w 07:1526, 07:1526
_cmd_grab_b_aux:NNw
..... 07:1526, 07:1527,
07:1529, 07:1531, 07:1533, 07:1534
_cmd_grab_b_end:Nw
..... 07:1526, 07:1537, 07:1542
_cmd_grab_b_long:w 07:1526, 07:1528
_cmd_grab_b_long_obey_spaces:w
..... 07:1526, 07:1532
_cmd_grab_b_obey_spaces:w ..
..... 07:1526, 07:1530
_cmd_grab_D:w 07:1553, 07:1553
_cmd_grab_D_aux:NNnNN . 07:1536,
07:1597, 07:1599, 07:1604, 07:1816
_cmd_grab_D_aux:NNnNNN
..... 07:1555, 07:1560,
07:1565, 07:1570, 07:1576, 07:1582,
07:1588, 07:1594, 07:1597, 07:1597
_cmd_grab_D_call:Nw
07:1601, 07:1661, 07:1661, 07:1818, 157
_cmd_grab_D_long:w 07:1553, 07:1558
_cmd_grab_D_long_no_strip:w ..
..... 07:1553, 07:1579
_cmd_grab_D_long_obey_spaces:w
..... 07:1553, 07:1568
_cmd_grab_D_long_obey_spaces_-
no_strip:w 07:1553, 07:1591
_cmd_grab_D_nested:NNnN
..... 07:1610, 07:1624, 07:1627
_cmd_grab_D_nested:w
..... 07:1624, 07:1642, 07:1659
_cmd_grab_D_no_strip:w ..
..... 07:1553, 07:1573
_cmd_grab_D_obey_spaces:w ..
..... 07:1553, 07:1563
_cmd_grab_D_obey_spaces_no_-
strip:w 07:1553, 07:1585
_cmd_grab_E:nnNN 07:1680, 07:1682,
07:1688, 07:1694, 07:1700, 07:1704
_cmd_grab_E:w 07:1680, 07:1680
_cmd_grab_E_finalise:
.. 07:1680, 07:1713, 07:1729, 07:1736
_cmd_grab_E_long:w 07:1680, 07:1686
_cmd_grab_E_long_obey_spaces:w
..... 07:1680, 07:1698
_cmd_grab_E_loop:NnN .. 07:1680,
07:1709, 07:1724, 07:1726, 07:1733
_cmd_grab_E_obey_spaces:w ..
..... 07:1680, 07:1692
_cmd_grab_expandably_bool ...
..... 07:31, 07:121,
07:448, 07:473, 07:475, 07:537,
```

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefsns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfnctcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltbypnem.dtx, 57=ltfinal.dtx

```

07:567, 07:592, 07:606, 07:628,
07:645, 07:707, 07:750, 07:807, 115
__cmd_grab_m:w 07:1737, 07:1737
__cmd_grab_m_1:w 07:1751
__cmd_grab_m_2:w 07:1751
__cmd_grab_m_3:w 07:1751
__cmd_grab_m_4:w 07:1751
__cmd_grab_m_5:w 07:1751
__cmd_grab_m_6:w 07:1751
__cmd_grab_m_7:w 07:1751
__cmd_grab_m_8:w 07:1751
__cmd_grab_m_9:w 07:1751
__cmd_grab_m_aux:Nnnnnnnn .. .
..... 07:1751, 07:1751,
07:1760, 07:1766, 07:1772, 07:1778,
07:1784, 07:1790, 07:1796, 07:1802
__cmd_grab_m_long:w 07:1737, 07:1744
__cmd_grab_R:w 07:1810, 07:1810
__cmd_grab_R_aux:NNnN .. .
... 07:1810, 07:1811, 07:1813, 07:1814
__cmd_grab_R_long:w 07:1810, 07:1812
__cmd_grab_t:w 07:1826, 07:1826
__cmd_grab_t_aux:NNw .. .
... 07:1826, 07:1827, 07:1829, 07:1830
__cmd_grab_t_obey_spaces:w .. .
..... 07:1826, 07:1828
__cmd_grab_v:w 07:1842, 07:1842
__cmd_grab_v_aux:w .. .
07:1842, 07:1845, 07:1850, 07:1852, 166
__cmd_grab_v_aux_abort:n .. .
07:1869, 07:1887, 07:1893, 07:1906,
07:1925, 07:1948, 07:1950, 07:1958, 165
__cmd_grab_v_aux_catcodes: .. .
07:1884, 07:1916, 07:1950, 07:1950, 164
__cmd_grab_v_aux_loop:N .. .
... 07:1879, 07:1885, 07:1889, 07:1903
__cmd_grab_v_aux_loop>NN .. .
... 07:1879, 07:1892, 07:1895
__cmd_grab_v_aux_loop_end: .. .
... 07:1879, 07:1900, 07:1908, 07:1939
__cmd_grab_v_aux_put:N .. .
..... 07:1883, 07:1902,
07:1918, 07:1936, 07:1944, 07:1978,
07:1978, 07:1980, 07:1994, 07:1996
__cmd_grab_v_aux_test:N .. .
..... 07:1868, 07:1879, 07:1879
__cmd_grab_v_bgroup: .. .
... 07:1864, 07:1914, 07:1914
__cmd_grab_v_bgroup_loop: .. .
..... 07:1914,
07:1919, 07:1921, 07:1937, 07:1945
__cmd_grab_v_bgroup_loop:N .. .
..... 07:1914, 07:1924, 07:1927
__cmd_grab_v_group_end:
07:1842, 07:1873, 07:1910, 07:1960, 164
__cmd_grab_v_long:w 07:1842, 07:1847
__cmd_grab_v_token_if_char:NTF .. .
..... 07:1881,
07:1897, 07:1929, 07:2005, 07:2005, 165
\g__cmd_grabber_int .. .
..... 07:27, 07:1075, 07:1079, 114
__cmd_if_recursion_tail_stop_- do:Nn . 07:46, 07:48, 07:2354, 07:2382
\c__cmd_ignore_def_t1 .. 07:2546,
07:2562, 07:2569, 07:2583, 07:2604,
07:2633, 07:2640, 07:2647, 07:2655,
07:2663, 07:2672, 07:2679, 07:2686,
07:2693, 07:2700, 07:2712, 07:2719, 183
\l__cmd_last_delimiters_t1 .. .
..... 07:33, 07:446,
07:465, 07:591, 07:605, 07:627,
07:663, 07:713, 07:723, 07:777, 127
\l__cmd_long_bool .. .
07:34, 07:450, 07:543, 07:544, 07:630,
07:741, 07:752, 07:757, 07:761,
07:783, 07:814, 07:931, 07:940,
07:976, 07:991, 07:1004, 07:1021,
07:1041, 07:1844, 07:1849, 07:1954, 115
\l__cmd_m_args_int .. .
..... 07:36, 07:786, 07:887,
07:915, 07:918, 07:920, 07:922, 115
\l__cmd_nesting_a_t1 .. .
07:1624
\l__cmd_nesting_b_t1 .. .
07:1624
__cmd_new_env:nnnn .. .
..... 07:2840, 07:2844,
07:2858, 07:2889, 07:2915, 07:2916
__cmd_normalize_arg_spec:n .. .
..... 07:89, 07:443, 07:443
__cmd_normalize_arg_spec_loop:n .. .
..... 07:443, 07:455,
07:477, 07:538, 07:583, 07:593,
07:608, 07:631, 07:637, 07:647, 07:653
__cmd_normalize_check_gv:N .. .
..... 07:651, 07:699, 07:699
__cmd_normalize_check_lu:N .. .
..... 07:699, 07:709
__cmd_normalize_E_unique_- check:w 07:585, 07:601, 07:610, 07:615
__cmd_normalize_type_!:_w .. 07:526
__cmd_normalize_type_+:_w .. 07:526
__cmd_normalize_type_=:_w .. 07:526
__cmd_normalize_type_>:_w .. 07:526
__cmd_normalize_type_aux:NnnN .. .
07:526, 07:542, 07:548, 07:557, 07:572
__cmd_normalize_type_b:w .. .
..... 07:655, 07:655

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__cmd_normalize_type_D:w
 07:506, 07:514, 07:516, 07:585, 07:585
__cmd_normalize_type_d:w
 07:501, 07:503
__cmd_normalize_type_E:w
 07:511, 07:585, 07:595
__cmd_normalize_type_e:w
 07:501, 07:508
__cmd_normalize_type_m:w
 07:633, 07:633
__cmd_normalize_type_0:w
 07:501, 07:515
__cmd_normalize_type_o:w
 07:501, 07:513
__cmd_normalize_type_R:w
 07:520, 07:633, 07:639
__cmd_normalize_type_r:w
 07:501, 07:517
__cmd_normalize_type_s:w
 07:501, 07:522
__cmd_normalize_type_t:w
 07:523, 07:585, 07:618
__cmd_normalize_type_v:w
 07:633, 07:649
\l__cmd_obey_spaces_bool .. 07:32,
 07:449, 07:549, 07:551, 07:624,
 07:629, 07:758, 07:762, 07:771,
 07:784, 07:821, 07:932, 07:941, 115
__cmd_peek_cs_check_equal:NNN ..
 07:2510, 07:2524, 07:2530
__cmd_peek_meaning:NTF
 07:2501, 07:2510, 07:2510
__cmd_peek_meaning_aux:NNTF ...
 .. 07:2510, 07:2511, 07:2513, 07:2514
__cmd_peek_meaning_remove:NTF ..
 07:1566, 07:1571,
 07:1589, 07:1595, 07:1696, 07:1702,
 07:1829, 07:2503, 07:2510, 07:2512
__cmd_peek_nonspace:NTF
 07:2500, 07:2500
__cmd_peek_nonspace_aux:nNNTF ..
 07:2500,
 07:2501, 07:2503, 07:2504, 07:2507
__cmd_peek_nonspace_remove:NTF ..
 07:1556, 07:1561,
 07:1577, 07:1583, 07:1684, 07:1690,
 07:1817, 07:1827, 07:2500, 07:2502
__cmd_peek_true_remove:NNw 07:2510
__cmd_peek_true_remove:Nw
 .. 07:2521, 07:2525, 07:2533, 07:2537
\l__cmd_prefixed_bool
 07:37, 07:798, 07:815,
 07:822, 07:828, 07:836, 07:885, 115
__cmd_prepare_signature:N
 07:780, 07:793, 07:796,
 07:848, 07:856, 07:869, 07:888,
 07:896, 07:904, 07:911, 07:977,
 07:987, 07:1029, 07:1042, 07:1055, 136
__cmd_prepare_signature:n
 07:90, 07:780, 07:780
__cmd_prepare_signature_-
 bypass:N 07:780, 07:799, 07:801,
 07:816, 07:823, 07:831, 07:841, 137
\l__cmd_process_all_tl
 ... 07:38, 07:174, 07:309, 07:317,
 07:382, 07:789, 07:919, 07:943, 125
\l__cmd_process_one_tl
 ... 07:39, 07:790, 07:830, 07:839,
 07:865, 07:874, 07:947, 07:950, 116
\l__cmd_process_some_bool . 07:40,
 07:173, 07:791, 07:829, 07:838, 116
__cmd_provide_env:nnnn
 07:2840, 07:2852,
 07:2883, 07:2891, 07:2919, 07:2920
__cmd_put_arg_expandable:nw ...
 07:2049,
 07:2054, 07:2055, 07:2090, 07:2095,
 07:2096, 07:2203, 07:2203, 07:2204
__cmd_renew_env:nnnn
 07:2840, 07:2848,
 07:2874, 07:2890, 07:2917, 07:2918
__cmd_replicate_processor:nn ...
 07:864, 07:871, 07:871
__cmd_run_code:
 07:311, 07:314, 07:314,
 07:1534, 07:1542, 07:1550, 07:1553,
 07:1558, 07:1563, 07:1568, 07:1574,
 07:1580, 07:1586, 07:1592, 07:1680,
 07:1686, 07:1692, 07:1698, 07:1722,
 07:1737, 07:1744, 07:1755, 07:1757,
 07:1763, 07:1769, 07:1775, 07:1781,
 07:1787, 07:1793, 07:1799, 07:1810,
 07:1812, 07:1830, 07:1852, 07:2010, 122
\l__cmd_saved_args_tl
 07:41, 07:1538, 07:1546, 116
__cmd_set_environment_end:n ...
 07:224, 07:275, 07:319
__cmd_set_eq_if_exist>NN
 07:1093, 07:1111, 07:1113,
 07:1132, 07:1151, 07:1152, 07:1153,
 07:1166, 07:1167, 07:1168, 07:1266
__cmd_show:N
 07:1310, 07:1312, 07:1312, 07:1339,
 07:1362, 07:1439, 07:1441, 07:1523, 153
__cmd_show:n 07:1329, 07:1425, 07:1431
__cmd_show_command:N
 07:1318, 07:1329, 07:1329

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__cmd_show_command:NnNNwN
..... 07:1329, 07:1330, 07:1331
__cmd_show_command_aux:NnNn ...
..... 07:1329, 07:1333,
07:1346, 07:1355, 07:1364, 07:1425, 153
__cmd_show_delim:Nw 07:1471, 07:1471
__cmd_show_delims:Nw 07:1471, 07:1473
__cmd_show_delims_opt:Nw
..... 07:1471, 07:1475
__cmd_show_E:Nw ... 07:1471, 07:1490
__cmd_show_e:Nw ... 07:1471, 07:1479
__cmd_show_environment:N
.. 07:1321, 07:1329, 07:1411, 07:1436
__cmd_show_environment:Nnnw ...
..... 07:1413, 07:1421
__cmd_show_environment_end:N ...
..... 07:1322, 07:1433
__cmd_show_expandable:N
..... 07:1319, 07:1329, 07:1336
__cmd_show_expandable:NnNNNNnN ...
..... 07:1329, 07:1337, 07:1342,
07:1344, 07:1351, 07:1353, 07:1359
__cmd_show_opt:Nw ... 07:1471, 07:1477
__cmd_show_optimized:N
..... 07:1320, 07:1329, 07:1374
__cmd_show_optimized:NN
..... 07:1376, 07:1380
__cmd_show_optimized_aux:N ...
..... 07:1382, 07:1405
__cmd_show_prefix:Nw 07:1471, 07:1503
__cmd_show_processor:Nw
..... 07:1471, 07:1505
\c__cmd_show_type_!_tl 07:1465
\c__cmd_show_type_+_tl 07:1465
\c__cmd_show_type_>_tl 07:1465
\c__cmd_show_type_D_tl 07:1465
\c__cmd_show_type_d_tl 07:1465
\c__cmd_show_type_E_tl 07:1465
\c__cmd_show_type_e_tl 07:1465
\c__cmd_show_type_0_tl 07:1465
\c__cmd_show_type_R_tl 07:1465
\c__cmd_show_type_r_tl 07:1465
\c__cmd_show_type_t_tl 07:1465
\l__cmd_signature_tl 07:42,
07:167, 07:210, 07:792, 07:855,
07:868, 07:895, 07:903, 07:917,
07:926, 07:1059, 07:1537, 07:1606,
07:1712, 07:1722, 07:1739, 07:1746,
07:1755, 07:1759, 07:1765, 07:1771,
07:1777, 07:1783, 07:1789, 07:1795,
07:1801, 07:1832, 07:1854, 07:2010, 116
__cmd_single_token_check:n ...
..... 07:588, 07:589, 07:599,
07:621, 07:642, 07:643, 07:671, 07:671
\l__cmd_some_long_bool
... 07:44, 07:114, 07:143, 07:204,
07:212, 07:453, 07:742, 07:753, 116
\l__cmd_some_obey_spaces_bool ...
... 07:43, 07:452, 07:552, 07:766, 116
\l__cmd_some_short_bool
..... 07:45, 07:113,
07:203, 07:214, 07:454, 07:754, 116
__cmd_split_add_item:n
07:1504, 07:1507, 07:1506, 07:1512, 155
__cmd_split_argument:nnn ...
..... 07:2246, 07:2246, 07:3002
__cmd_split_argument_aux:n ...
..... 07:2246, 07:2262, 07:2281
__cmd_split_argument_aux:nnn ...
..... 07:2246, 07:2249, 07:2253
__cmd_split_argument_aux:wn ...
..... 07:2246, 07:2282, 07:2283
__cmd_split_end_item:n . 07:1463,
07:1472, 07:1474, 07:1476, 07:1478,
07:1484, 07:1495, 07:1507, 07:1514, 155
__cmd_split_list:nn
.. 07:2211, 07:2213, 07:2248, 07:3003
__cmd_split_list_multi:nn ...
..... 07:2211, 07:2218,
07:2221, 07:2223, 07:2230, 07:2243
\l__cmd_split_list_seq
..... 07:2211, 07:2225, 07:2227, 175
__cmd_split_list_single:Nn ...
..... 07:2211, 07:2219, 07:2233
\l__cmd_split_list_tl
07:2212, 07:2235, 07:2241, 07:2243, 175
__cmd_split_N_head_apply:Nn ...
.. 07:2291, 07:2311, 07:2323, 07:2398
__cmd_split_N_head_apply_- aux:NNw ... 07:2291, 07:2399, 07:2400
__cmd_split_signature:n
..... 07:1366, 07:1442, 07:1442, 152
__cmd_split_signature_loop:Nw ...
07:1447, 07:1449, 07:1449, 07:1463,
07:1472, 07:1474, 07:1476, 07:1478,
07:1486, 07:1499, 07:1504, 07:1506
__cmd_split_start_item: 07:1452,
07:1483, 07:1494, 07:1507, 07:1507, 154
__cmd_start:nNnnn
.. 07:163, 07:283, 07:294, 07:2481, 138
__cmd_start_aux:NNnnnn
.. 07:289, 07:300, 07:303, 07:303, 07:313
__cmd_start_env:nnnnn
.. 07:159, 07:283, 07:283, 07:2484, 181
__cmd_start_expandable:nNNNNn ...
.. 07:197, 07:401, 07:401, 07:2482, 146
__cmd_start_optimized: .. 07:106,
07:135, 07:149, 07:1185, 07:2483, 181

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\l__cmd_suppress_strip_bool 45:507, 54:80, 54:146, 54:147,
... 07:35, 07:451, 07:558, 07:566, 54:148, 54:203, 54:204, 54:205,
07:785, 07:837, 07:934, 07:942, 115 54:206, 54:207, 54:1847, 54:1849,
__cmd_t1_mapthread_function:NNN 54:2262, 54:2266, 54:2294, 54:2300
.... 07:352, 07:380, 07:2447, 07:2447 \cong 30:464
__cmd_t1_mapthread_function:nnN ... 44:164, 44:171, 44:177, 44:184, 908
.... 07:428, 07:1497, 07:2447, 07:2460, 155 \coprod 30:348
__cmd_t1_mapthread_loop:w \copyright ... 21:285, 21:316, 29:649, 1218
.... 07:2447, 07:2450, 07:2462, 07:2466, 07:2471 \coremissesfalse 33:1256
__cmd_tmp:w 07:49, \coremissestrue 33:1234
.... 07:52, 07:338, 07:354, 07:396, \cos 38:12
07:398, 07:501, 07:525, 07:1064, \cosh 38:14
07:1070, 07:1088, 07:1295, 07:1305, \cot 38:18
07:1465, 07:1468, 07:1470, 07:1488, \coth 38:19
07:1492, 07:1498, 07:1502, 07:2015, \countdef 04:75, 04:85,
07:2034, 07:2067, 07:2086, 07:2143, 04:174, 04:190, 04:198, 04:206,
07:2167, 07:2170, 07:2194, 07:2542, 169 04:225, 34:3, 02:37, 57:75, 02:38,
02:39, 02:41, 02:51, 02:90, 01:50
\l__cmd_tmp_prop counter 36:242, 751
.... 07:49, 07:1708, 07:1711, 07:1717, 117 \counterwithin 512
\l__cmd_tmpt1 ... 07:50, 07:350, \counterwithin 22:123, 1258
07:355, 07:365, 07:1033, 07:1035, \counterwithout 512
07:1049, 07:1052, 07:1206, 07:1212, \counterwithout 22:123, 1258
07:1227, 07:1235, 07:1251, 07:1258, \CountZero 04:225
07:1276, 07:1279, 07:1370, 07:1435, \cr 21:500, 21:506, 21:516, 21:522, 33:101,
07:1436, 07:1445, 07:1516, 07:1645, 33:105, 33:927, 33:931, 38:188,
07:1648, 07:2516, 07:2541, 07:2544, 117 38:192, 38:378, 38:424, 38:540,
\l__cmd_tmptb_t1 ... 07:51, 07:1018, 41:192, 41:203, 41:210, 41:219,
07:1023, 07:1034, 07:1277, 07:1279, 41:224, 41:230, 41:377, 42:141,
07:1446, 07:1452, 07:1509, 07:1513, 42:143, 42:148, 42:154, 02:446, 1166
07:1517, 07:1518, 07:1717, 07:1718, \crcr 21:327, 21:362, 21:363,
07:1720, 07:2517, 07:2528, 07:2534, 154 21:390, 21:394, 21:397, 21:476,
__cmd_token_if_cs:NTF 21:480, 21:484, 21:486, 21:489,
07:1669, 07:2438, 07:2438, 07:2523, 133 21:732, 21:760, 21:764, 21:767,
__cmd_trim_spaces:n 21:834, 21:837, 21:895, 21:1253,
.... 07:2289, 07:2289, 07:3004 29:651, 30:344, 30:345, 30:347,
__cmd_use_i_delimit_by_q_- 30:466, 30:469, 30:473, 30:537,
recursion_stop:nw . 07:46, 07:2358 30:538, 30:539, 30:540, 30:541,
\l__cmd_v_arg_t1 30:542, 30:544, 30:545, 30:546,
07:1841, 07:1858, 07:1877, 07:1911, 30:547, 30:548, 30:550, 33:106,
07:1966, 07:1973, 07:1982, 07:1998, 164 33:932, 38:168, 38:170, 38:171,
\l__cmd_v_nesting_int .. 07:1913, 38:172, 38:188, 38:190, 38:191,
07:1917, 07:1933, 07:1934, 07:1943, 166 38:192, 38:210, 38:211, 41:171,
\cmrFont 33:1249, 33:1252 41:172, 42:141, 55:237, 02:536, 1171
\colon 30:510
\color 410
\columnbreak 952
\columnsep 20:27, 20:97, 20:154, 54:81, 54:204
\columnseprule ... 54:82, 54:2264, 54:2298
\columnwidth 20:24, 20:27, 20:28,
20:30, 20:94, 20:97, 20:98, 20:100,
20:151, 20:154, 20:155, 20:158,
40:444, 40:483, 40:501, 40:518,
45:99, 45:168, 45:470, 45:489,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

28:322, 51:180, 52:211, 52:225,  
 52:237, 52:238, 07:1672, 07:2487,  
 08:381, 08:1066, 08:1123, 08:1274,  
 08:1414, 08:1521, 08:1522, 08:1583,  
 08:1603, 08:1606, 08:1667, 08:1692,  
 08:1695, 08:1713, 08:1714, 08:1738,  
 08:1739, 08:1740, 08:1747, 08:1754, 236  
`\cs_argument_spec:N` . . . . . 05:141, 1268  
`\cs_end:` . . . . . 08:2099, 08:2205,  
 08:2209, 08:2213, 08:2225, 08:2230,  
 08:2231, 08:2245, 08:2251, 08:2265,  
 08:2278, 08:2283, 08:2331, 08:2417,  
 08:2418, 08:2483, 08:2533, 08:2916,  
 08:2936, 08:2946, 28:319, 28:322,  
 51:180, 52:211, 52:225, 52:237,  
 52:238, 07:1672, 07:2489, 08:381,  
 08:767, 08:1066, 08:1123, 08:1277,  
 08:1417, 08:1521, 08:1522, 08:1583,  
 08:1603, 08:1606, 08:1667, 08:1692,  
 08:1695, 08:1713, 08:1714, 08:1729,  
 08:1739, 08:1740, 08:1747, 08:1754  
`\cs_generate_from_arg_count>NNnn`  
 . . . . . 10:81,  
 10:100, 11:397, 11:459, 11:818,  
 07:138, 07:152, 07:180, 07:188,  
 07:268, 07:354, 07:357, 07:396, 1266  
`\cs_generate_variant:Nn` . . . 09:20,  
 11:64, 11:349, 11:502, 11:703,  
 11:734, 11:807, 20:528, 36:37, 36:52,  
 36:62, 36:79, 36:96, 36:116, 36:117,  
 36:230, 48:264, 51:51, 51:52, 51:53,  
 51:177, 52:461, 52:469, 57:596,  
 07:236, 07:313, 07:357, 07:1113,  
 07:1410, 07:2012, 07:2204, 07:2230,  
 07:2889, 07:2890, 07:2891, 08:37,  
 08:38, 08:39, 08:40, 08:41, 08:42,  
 08:43, 08:53, 08:54, 08:57, 08:66,  
 08:1080, 08:1103, 08:1633, 08:1716  
`\cs_gset:Npn` . . . . .  
 . 08:2288, 08:2358, 11:811, 28:285,  
 28:324, 36:24, 52:242, 53:173,  
 08:1079, 08:1099, 08:1629, 08:2060  
`\cs_gset:Npx` . . . . . 52:213, 08:1047  
`\cs_gset_eq:NN` . . . . .  
 . . . . . 08:2194, 08:2295, 08:2296,  
 08:2297, 08:2298, 08:2562, 09:56,  
 09:539, 09:567, 11:642, 28:283,  
 28:291, 28:316, 28:330, 05:136,  
 05:137, 05:138, 05:140, 05:141,  
 05:142, 05:143, 05:144, 05:145,  
 53:52, 53:151, 53:166, 53:167,  
 53:172, 53:181, 53:185, 53:321,  
 53:387, 07:3006, 08:770, 08:1424,  
 08:1450, 08:1465, 08:1466, 08:1469  
`\cs_gset_nopar:Npn` . . . . . 08:2944  
`\cs_gset_nopar:Npx` . . . . . 08:50, 08:52  
`\cs_gset_protected:Npn` . . . . .  
 . . . . . 08:2139, 08:2142,  
 08:2170, 08:2175, 08:2193, 08:2352,  
 08:2382, 08:2778, 08:2921, 09:231,  
 09:423, 09:545, 11:399, 11:828,  
 51:62, 08:88, 08:91, 08:100, 08:133,  
 08:160, 08:188, 08:191, 08:197,  
 08:221, 08:226, 08:251, 08:310,  
 08:328, 08:581, 08:584, 08:602,  
 08:628, 08:677, 08:685, 08:1448, 08:1509  
`\cs_gset_protected:Npx` . . . . . 10:21,  
 10:23, 48:318, 53:20, 08:20, 08:1161  
`\cs_if_eq:NNTF` . . . . .  
 51:97, 51:99, 51:165, 07:1070, 07:1728  
`\cs_if_exist:N` . . . . . 1032  
`\cs_if_exist:NTF` . . . . .  
 . 08:2464, 08:2493, 08:2501, 09:73,  
 09:81, 10:76, 10:98, 10:119, 11:54,  
 11:60, 11:73, 11:86, 11:395, 11:453,  
 11:488, 11:640, 11:663, 11:679,  
 36:67, 36:157, 05:139, 51:167, 52:93,  
 52:438, 52:442, 55:8, 07:67, 07:229,  
 07:245, 07:1073, 07:1112, 07:2808,  
 07:2821, 07:2833, 07:2860, 07:2866,  
 07:2876, 07:2885, 07:2897, 07:2900,  
 07:2907, 07:2912, 07:2927, 07:2940,  
 07:2953, 08:1051, 08:1116, 1053  
`\cs_if_exist_p:N` . 07:74, 07:75, 08:277  
`\cs_if_exist_use:NTF` . . . . .  
 . . . . . 11:728, 05:184, 53:313,  
 53:314, 53:318, 53:319, 07:481,  
 07:1229, 08:1283, 08:1327, 08:1352  
`\cs_if_free:NTF` . . . . . 36:8, 51:106  
`\cs_meaning:N` . . . . . 10:64  
`\cs_new:Npn` . . . . . 08:2203,  
 08:2228, 08:2236, 08:2249, 08:2257,  
 08:2266, 08:2275, 08:2310, 08:2491,  
 08:2499, 08:2513, 08:2531, 08:2540,  
 08:2545, 08:2786, 08:2787, 08:2788,  
 08:2789, 08:2793, 08:2794, 09:303,  
 09:313, 09:323, 09:325, 09:327,  
 09:338, 09:361, 09:362, 09:365,  
 09:485, 09:593, 10:83, 10:142,  
 11:350, 11:351, 11:454, 11:508,  
 11:908, 11:1186, 11:1188, 11:1190,  
 16:85, 16:88, 16:113, 16:125, 20:519,  
 36:63, 36:65, 36:89, 36:97, 36:104,  
 36:167, 36:168, 36:182, 36:183,  
 36:197, 36:198, 05:182, 48:250,  
 48:260, 48:261, 48:262, 48:263,  
 51:174, 51:176, 52:30, 52:38, 52:44,  
 52:57, 52:95, 52:100, 52:105, 52:112,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

52:127, 52:234, 52:240, 52:411,  
 52:413, 52:415, 52:418, 52:422,  
 52:436, 52:452, 52:462, 52:582,  
 53:54, 53:59, 53:77, 53:141, 53:146,  
 53:163, 53:178, 53:183, 53:189,  
 53:192, 53:206, 53:257, 53:310,  
 53:323, 53:337, 53:340, 53:350,  
 53:392, 53:511, 53:512, 53:513,  
 55:14, 55:28, 55:128, 55:134, 55:186,  
 55:189, 55:232, 07:96, 07:98, 07:149,  
 07:401, 07:406, 07:408, 07:410,  
 07:422, 07:434, 07:440, 07:1136,  
 07:1191, 07:1272, 07:1297, 07:1302,  
 07:1659, 07:2013, 07:2017, 07:2035,  
 07:2042, 07:2043, 07:2065, 07:2069,  
 07:2087, 07:2100, 07:2102, 07:2104,  
 07:2106, 07:2112, 07:2129, 07:2131,  
 07:2133, 07:2135, 07:2137, 07:2139,  
 07:2141, 07:2145, 07:2168, 07:2172,  
 07:2195, 07:2197, 07:2203, 07:2281,  
 07:2283, 07:2308, 07:2314, 07:2320,  
 07:2326, 07:2398, 07:2400, 07:2447,  
 07:2460, 07:2466, 07:2548, 07:2962,  
 07:2979, 07:2980, 07:2984, 07:2985,  
 07:2986, 08:45, 08:46, 08:330,  
 08:336, 08:351, 08:361, 08:362,  
 08:364, 08:378, 08:384, 08:904,  
 08:906, 08:908, 08:1202, 08:1214,  
 08:1219, 08:1227, 08:1236, 08:1238,  
 08:1245, 08:1272, 08:1279, 08:1281,  
 08:1396, 08:1412, 08:1546, 08:1547,  
 08:1718, 08:1719, 08:2056, 08:2063  
  
\cs\_new\_eq:NN . . . . . 08:2265,  
08:2807, 08:2808, 08:2809, 08:2810,  
08:2880, 08:2882, 09:13, 09:14,  
09:440, 10:7, 10:8, 10:138, 10:139,  
10:140, 10:141, 10:145, 10:170,  
10:171, 10:172, 10:173, 10:174,  
10:175, 10:176, 10:177, 11:356,  
11:357, 11:358, 11:373, 16:135,  
20:797, 20:798, 36:166, 36:181,  
36:196, 05:162, 05:163, 05:164,  
05:165, 05:181, 48:305, 48:321,  
48:322, 48:363, 48:365, 48:475,  
48:477, 51:8, 52:252, 52:253, 52:509,  
52:511, 52:513, 52:515, 52:517,  
52:519, 52:521, 52:523, 52:525,  
52:527, 53:7, 53:152, 53:347, 53:349,  
53:411, 53:414, 53:415, 53:510, 55:3,  
55:4, 55:123, 57:624, 57:625, 07:52,  
07:95, 07:1247, 07:1248, 07:1255,  
07:1262, 07:2802, 07:2803, 07:2981,  
07:2982, 07:2983, 07:2989, 07:2990,  
07:2991, 07:3001, 07:3002, 07:3003,  
  
07:3004, 07:3005, 08:7, 08:23,  
08:34, 08:63, 08:1186, 08:1369,  
08:1376, 08:1419, 08:1793, 08:1794,  
08:1795, 08:1796, 08:1797, 08:1798  
\cs\_new\_protected:Npe . . . . . 11:275  
\cs\_new\_protected:Npn . . 08:2074,  
08:2092, 08:2097, 08:2106, 08:2124,  
08:2130, 08:2144, 08:2157, 08:2184,  
08:2188, 08:2198, 08:2211, 08:2220,  
08:2234, 08:2240, 08:2263, 08:2270,  
08:2293, 08:2316, 08:2326, 08:2337,  
08:2343, 08:2363, 08:2393, 08:2395,  
08:2405, 08:2547, 08:2553, 08:2559,  
08:2730, 08:2731, 08:2732, 08:2755,  
08:2766, 08:2796, 08:2797, 08:2798,  
08:2799, 08:2811, 08:2818, 08:2825,  
08:2832, 08:2839, 08:2841, 08:2843,  
08:2850, 08:2857, 08:2864, 08:2871,  
09:18, 09:23, 09:25, 09:34, 09:36,  
09:46, 09:54, 09:63, 09:77, 09:97,  
09:117, 09:130, 09:135, 09:144,  
09:149, 09:156, 09:296, 09:367,  
09:404, 09:441, 09:469, 09:498, 10:9,  
10:14, 10:19, 10:27, 10:53, 10:72,  
10:73, 10:95, 10:116, 10:132, 11:42,  
11:52, 11:58, 11:65, 11:71, 11:96,  
11:104, 11:120, 11:128, 11:136,  
11:146, 11:162, 11:172, 11:182,  
11:188, 11:203, 11:219, 11:241,  
11:253, 11:270, 11:296, 11:320,  
11:352, 11:354, 11:359, 11:361,  
11:363, 11:365, 11:367, 11:369,  
11:371, 11:374, 11:393, 11:401,  
11:412, 11:414, 11:428, 11:432,  
11:446, 11:503, 11:522, 11:540,  
11:561, 11:578, 11:585, 11:593,  
11:598, 11:607, 11:613, 11:618,  
11:627, 11:632, 11:645, 11:654,  
11:673, 11:686, 11:698, 11:704,  
11:710, 11:715, 11:735, 11:743,  
11:748, 11:759, 11:765, 11:785,  
11:802, 11:808, 11:814, 11:825,  
11:831, 11:843, 11:853, 11:872,  
11:874, 11:885, 11:890, 11:896,  
11:902, 11:909, 11:911, 11:913,  
11:922, 11:931, 11:940, 11:948,  
11:1156, 11:1158, 11:1160, 11:1162,  
11:1164, 11:1166, 11:1168, 11:1170,  
11:1172, 11:1174, 11:1176, 11:1178,  
11:1180, 11:1182, 11:1184, 11:1192,  
11:1193, 11:1194, 16:93, 28:278,  
36:6, 36:17, 36:22, 36:38, 36:43,  
36:48, 36:50, 36:53, 36:128, 36:142,  
36:149, 36:199, 36:206, 36:214, 48:7,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

48:17, 48:65, 48:73, 48:117, 48:122,  
 48:147, 48:156, 48:194, 48:205,  
 48:213, 48:216, 48:248, 48:306,  
 48:311, 48:316, 48:324, 48:352,  
 48:358, 48:381, 48:412, 51:22,  
 51:28, 51:44, 51:58, 51:60, 51:95,  
 51:104, 51:116, 51:132, 51:137,  
 51:146, 51:158, 51:163, 51:178,  
 51:182, 51:200, 51:214, 52:49,  
 52:62, 52:70, 52:79, 52:208, 52:222,  
 52:392, 52:397, 52:420, 52:547, 53:8,  
 53:13, 53:18, 53:343, 53:473, 55:5,  
 55:6, 55:15, 55:16, 55:17, 55:18,  
 55:124, 55:155, 55:158, 55:178,  
 55:182, 55:192, 55:195, 55:198,  
 55:201, 55:204, 55:221, 57:577,  
 57:605, 57:610, 57:615, 57:623,  
 07:55, 07:60, 07:65, 07:86, 07:106,  
 07:128, 07:150, 07:178, 07:226,  
 07:240, 07:261, 07:275, 07:283,  
 07:303, 07:314, 07:324, 07:335,  
 07:343, 07:348, 07:358, 07:363,  
 07:377, 07:386, 07:394, 07:443,  
 07:477, 07:503, 07:508, 07:513,  
 07:515, 07:517, 07:522, 07:526,  
 07:540, 07:546, 07:555, 07:572,  
 07:585, 07:595, 07:610, 07:618,  
 07:633, 07:639, 07:649, 07:655,  
 07:671, 07:680, 07:699, 07:709,  
 07:721, 07:732, 07:737, 07:738,  
 07:764, 07:780, 07:796, 07:801,  
 07:811, 07:818, 07:825, 07:833,  
 07:843, 07:850, 07:858, 07:871,  
 07:882, 07:890, 07:898, 07:906,  
 07:913, 07:924, 07:952, 07:962,  
 07:967, 07:974, 07:979, 07:981,  
 07:989, 07:1002, 07:1015, 07:1031,  
 07:1037, 07:1044, 07:1046, 07:1057,  
 07:1062, 07:1068, 07:1086, 07:1093,  
 07:1111, 07:1114, 07:1129, 07:1148,  
 07:1163, 07:1177, 07:1208, 07:1216,  
 07:1224, 07:1233, 07:1245, 07:1249,  
 07:1256, 07:1261, 07:1263, 07:1274,  
 07:1281, 07:1289, 07:1312, 07:1329,  
 07:1331, 07:1336, 07:1344, 07:1353,  
 07:1364, 07:1374, 07:1380, 07:1411,  
 07:1421, 07:1431, 07:1433, 07:1442,  
 07:1449, 07:1471, 07:1473, 07:1475,  
 07:1477, 07:1479, 07:1490, 07:1503,  
 07:1505, 07:1507, 07:1512, 07:1514,  
 07:1526, 07:1528, 07:1530, 07:1532,  
 07:1534, 07:1542, 07:1553, 07:1558,  
 07:1563, 07:1568, 07:1573, 07:1579,  
 07:1585, 07:1591, 07:1597, 07:1604,  
 07:1627, 07:1680, 07:1686, 07:1692,  
 07:1698, 07:1704, 07:1726, 07:1736,  
 07:1737, 07:1744, 07:1757, 07:1763,  
 07:1769, 07:1775, 07:1781, 07:1787,  
 07:1793, 07:1799, 07:1810, 07:1812,  
 07:1814, 07:1826, 07:1828, 07:1830,  
 07:1842, 07:1847, 07:1852, 07:1873,  
 07:1879, 07:1889, 07:1895, 07:1908,  
 07:1921, 07:1927, 07:1950, 07:1958,  
 07:1980, 07:1996, 07:2005, 07:2007,  
 07:2205, 07:2213, 07:2223, 07:2233,  
 07:2246, 07:2253, 07:2289, 07:2291,  
 07:2296, 07:2332, 07:2337, 07:2348,  
 07:2350, 07:2352, 07:2369, 07:2380,  
 07:2390, 07:2392, 07:2394, 07:2396,  
 07:2406, 07:2410, 07:2438, 07:2473,  
 07:2494, 07:2500, 07:2502, 07:2504,  
 07:2510, 07:2512, 07:2514, 07:2530,  
 07:2537, 07:2804, 07:2817, 07:2830,  
 07:2835, 07:2842, 07:2846, 07:2850,  
 07:2854, 07:2858, 07:2874, 07:2883,  
 07:2895, 07:2905, 07:2911, 07:2913,  
 07:2923, 07:2936, 07:2949, 07:2957,  
 08:8, 08:13, 08:18, 08:47, 08:49,  
 08:51, 08:55, 08:58, 08:64, 08:69,  
 08:71, 08:73, 08:104, 08:148, 08:172,  
 08:174, 08:176, 08:201, 08:203,  
 08:205, 08:231, 08:266, 08:268,  
 08:285, 08:290, 08:292, 08:385,  
 08:394, 08:399, 08:407, 08:431,  
 08:433, 08:449, 08:457, 08:467,  
 08:477, 08:488, 08:494, 08:500,  
 08:516, 08:529, 08:555, 08:572,  
 08:632, 08:651, 08:659, 08:666,  
 08:698, 08:704, 08:710, 08:720,  
 08:762, 08:764, 08:772, 08:774,  
 08:777, 08:779, 08:960, 08:962,  
 08:997, 08:1034, 08:1046, 08:1049,  
 08:1078, 08:1091, 08:1093, 08:1095,  
 08:1097, 08:1114, 08:1138, 08:1145,  
 08:1303, 08:1310, 08:1341, 08:1363,  
 08:1370, 08:1377, 08:1383, 08:1388,  
 08:1393, 08:1394, 08:1422, 08:1475,  
 08:1555, 08:1637, 08:1720, 08:1727,  
 08:1736, 08:1743, 08:1750, 08:1757,  
 08:1765, 08:1771, 08:1782, 08:1799,  
 08:1806, 08:1818, 08:1823, 08:1828,  
 08:1830, 08:1834, 08:1952, 1082  
 \cs\_new\_protected:Npx . . . 09:475,  
 48:100, 07:294, 07:1805, 07:1914  
 \cs\_new\_protected\_nopar:Npn . . . . .  
 . . . . . 57:620, 07:1661, 07:1751  
 \cs\_parameter\_spec:N . . . . .  
 05:139, 05:140, 07:1402, 08:1230, 1268

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\cs_prefix_spec:N 09:207, 09:282, 05:138
\cs_replacement_spec 253
\cs_replacement_spec:N 05:143, 07:1371, 07:1407, 07:1417,
08:37, 08:41, 08:1205, 08:1440, 08:1462, 08:1876, 08:1889, 08:1898
\cs_set:Npe 07:1382
\cs_set:Npn 08:2551, 08:2557, 08:2930, 09:199, 09:274, 09:456, 09:515, 09:551, 10:102, 11:328, 11:812, 16:37, 16:68, 07:190, 07:269, 07:354, 07:396, 07:1064, 08:1147, 08:1158, 08:1165, 08:1174, 08:1873, 156
\cs_set:Npx 52:211, 52:225, 07:219, 07:992, 07:1005
\cs_set_eq:NN 09:171, 09:172, 09:192, 09:193, 09:198, 09:246, 09:247, 09:267, 09:268, 09:273, 09:452, 09:453, 09:511, 09:512, 09:547, 09:548, 09:606, 11:681, 16:91, 16:92, 16:136, 16:137, 16:138, 16:140, 16:141, 16:142, 16:176, 16:177, 16:178, 16:181, 28:294, 28:333, 36:231, 48:99, 48:251, 48:252, 48:253, 51:85, 53:24, 53:44, 53:83, 53:95, 53:125, 53:132, 53:417, 53:419, 53:421, 53:423, 53:425, 55:118, 55:121, 07:271, 07:272, 07:1088, 07:1112, 07:1131, 07:1150, 07:1165, 07:1179, 07:1240, 07:1265, 07:1270, 07:1285, 07:1287, 07:1760, 07:1766, 07:1772, 07:1778, 07:1784, 07:1790, 07:1796, 07:1802, 07:1952, 07:2542, 08:1494, 08:1495, 08:1496, 08:1497, 08:1527, 08:1529, 08:1531, 08:1533, 08:1820, 08:1825
\cs_set_nopar:Npe 07:131, 07:1182
\cs_set_nopar:Npn 57:579, 156
\cs_set_nopar:Npx 07:195, 07:216, 07:221, 07:266, 07:277, 07:993, 07:1006, 07:1156, 07:1170, 07:1283, 08:48
\cs_set_protected:Npe 11:665
\cs_set_protected:Npn .. 08:2078, 08:2109, 09:173, 09:248, 11:324, 11:829, 28:311, 28:343, 51:5, 51:64, 51:78, 51:188, 53:46, 53:135, 53:412, 57:594, 07:154, 07:501, 07:1295, 07:1465, 07:1488, 07:1492, 07:1529, 07:1533, 07:1560, 07:1570, 07:1582, 07:1594, 07:1689, 07:1701, 07:1713, 07:1747, 07:1813, 07:1833, 07:2015, 07:2067, 07:2143, 07:2170, 08:420
\cs_set_protected_nopar:Npe
\cs_set_protected_nopar:Npn 07:132, 07:1181
\cs_set_protected_nopar:Npx 07:1527, 07:1531, 07:1555, 07:1565, 07:1576, 07:1588, 07:1683, 07:1695, 07:1740, 07:1811
\cs_set_protected_nopar:Npx 07:155, 07:195, 07:1133, 07:1156, 07:1268
\cs_show:N 11:912, 375
\cs_to_str:N 09:112, 09:133, 09:147, 09:152, 05:136, 07:74, 07:75, 07:88, 07:1105, 07:1107, 07:1325, 07:1377, 07:1416, 07:1418, 07:1435, 07:1672, 07:2489, 236
\cs_undefine:N 08:2303, 08:2389, 08:2527, 09:59, 09:210, 09:494, 09:502, 52:227, 07:2915, 07:2916, 07:2917, 07:2918, 07:2919, 07:2920, 07:2995, 07:2996, 07:2997, 07:3007, 08:79, 08:80, 08:90, 08:132, 08:190, 08:196, 08:271, 08:649, 08:1084, 08:1085, 08:1086, 08:1087, 08:1107, 08:1108, 08:1109, 08:1110, 08:1134, 08:1190, 08:1210, 08:1268, 08:1289, 08:1290, 08:1301, 08:1395
cs\check@cr commands:
\cs_gset_eq:NN 76
\csc 38:21
\csname 200
\csname\endcsname 1051
\cup 30:380
\CurrentFile 50:938, 52:3, 52:67, 52:68, 52:84, 52:166, 52:171, 52:174, 52:377, 52:381, 52:555, 52:556, 1046
\CurrentFilePath 52:3, 52:67, 52:83, 52:165, 52:378, 52:381, 1052
\CurrentFilePathUsed 52:3, 52:66, 52:81, 52:163, 52:375, 52:378, 1037
\CurrentFileUsed 50:938, 52:3, 52:66, 52:82, 52:164, 52:375, 52:377, 52:554, 52:555, 1037
\currentgrouplevel 39:148, 50:780
\CurrentOption 21:1536, 21:1539, 21:1544, 21:1560, 50:14, 50:520, 50:531, 50:544, 50:545, 50:550, 50:563, 50:564, 50:566, 50:580, 50:581, 50:584, 50:598, 50:603, 50:604, 50:617, 50:622, 50:623, 50:636, 50:638, 50:644, 50:646, 50:658, 50:659, 50:660, 50:668, 50:669, 50:670, 50:988, 50:1053, 50:1172, 50:1173, 50:1183, 51:82, 51:83, 1200

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=ltphypen.dtx, 57=ltfinal.dtx

|                                                                         |                  |
|-------------------------------------------------------------------------|------------------|
| CurrentOption commands:                                                 |                  |
| \CurrentOption: . . . . .                                               | 50:543, 21:1514  |
| 50:562, 50:579, 50:597, 50:602, 50:616, 50:621, 50:657, 50:667, 50:1182 | 21:1514          |
| \currsubencoding . . . . .                                              | 33:1248, 21:1508 |
| 33:1259, 33:1415, 33:1417, 33:1418                                      | 21:1514          |
| \CYRA . . . . .                                                         | 21:1508          |
| \cyra . . . . .                                                         | 21:1508, 21:1563 |
| \CYRABHCH . . . . .                                                     | 21:1508          |
| \cyrabhch . . . . .                                                     | 21:1508          |
| \CYRABHCHDSC . . . . .                                                  | 21:1508          |
| \cyrabhchdsc . . . . .                                                  | 21:1508          |
| \CYRABHDZE . . . . .                                                    | 21:1509          |
| \cyrabhdze . . . . .                                                    | 21:1508          |
| \CYRABHHA . . . . .                                                     | 21:1509          |
| \cyrabhha . . . . .                                                     | 21:1509          |
| \CYRAE . . . . .                                                        | 21:1509          |
| \cyrae . . . . .                                                        | 21:1509          |
| \CYRB . . . . .                                                         | 21:1509          |
| \cyrb . . . . .                                                         | 21:1509          |
| \CYRBYUS . . . . .                                                      | 21:1510          |
| \cyrbyus . . . . .                                                      | 21:1509          |
| \CYRC . . . . .                                                         | 21:1510          |
| \cyrc . . . . .                                                         | 21:1510          |
| \CYRCH . . . . .                                                        | 21:1510          |
| \cyrch . . . . .                                                        | 21:1510          |
| \CYRCHLDSC . . . . .                                                    | 21:1510          |
| \cyrchldsc . . . . .                                                    | 21:1510          |
| \CYRCHRDSC . . . . .                                                    | 21:1511          |
| \cyrchrdsc . . . . .                                                    | 21:1510          |
| \CYRCHVCRS . . . . .                                                    | 21:1511          |
| \cyrchvcrs . . . . .                                                    | 21:1511          |
| \CYRD . . . . .                                                         | 21:1511          |
| \cyrd . . . . .                                                         | 21:1511          |
| \CYRDELTA . . . . .                                                     | 21:1511          |
| \cyrdelta . . . . .                                                     | 21:1511          |
| \CYRDJE . . . . .                                                       | 21:1512          |
| \cyrdje . . . . .                                                       | 21:1512          |
| \CYRDZE . . . . .                                                       | 21:1512          |
| \cyrdze . . . . .                                                       | 21:1512          |
| \CYRDZHE . . . . .                                                      | 21:1512          |
| \cyrdzhe . . . . .                                                      | 21:1512          |
| \CYRE . . . . .                                                         | 21:1512          |
| \cyre . . . . .                                                         | 21:1512          |
| \CYREPS . . . . .                                                       | 21:1513          |
| \cyreps . . . . .                                                       | 21:1512          |
| \CYREREV . . . . .                                                      | 21:1513          |
| \cyrerev . . . . .                                                      | 21:1513          |
| \CYRERY . . . . .                                                       | 21:1513          |
| \cyrery . . . . .                                                       | 21:1513          |
| \CYRF . . . . .                                                         | 21:1513          |
| \cyrf . . . . .                                                         | 21:1513          |
| \CYRFITA . . . . .                                                      | 21:1514          |
| \cyrfita . . . . .                                                      | 21:1513          |
| \CYRG . . . . .                                                         | 21:1514          |
| \cyrg . . . . .                                                         | 21:1514          |
| \CYRGDSC . . . . .                                                      | 21:1514          |
| \cyrgdsc . . . . .                                                      | 21:1514          |
| \CYRGDSCHCRS . . . . .                                                  | 21:1514          |
| \cyrghcrs . . . . .                                                     | 21:1514          |
| \CYRGHCRS . . . . .                                                     | 21:1515          |
| \cyrghcrs . . . . .                                                     | 21:1515          |
| \CYRGHK . . . . .                                                       | 21:1515          |
| \cyrghk . . . . .                                                       | 21:1515          |
| \CYRGUP . . . . .                                                       | 21:1515          |
| \cyrup . . . . .                                                        | 21:1515          |
| \CYRH . . . . .                                                         | 21:1515          |
| \cyrh . . . . .                                                         | 21:1515          |
| \CYRHDSC . . . . .                                                      | 21:1516          |
| \cyrhdsc . . . . .                                                      | 21:1516          |
| \CYRHHCRS . . . . .                                                     | 21:1516          |
| \cyrhhcrs . . . . .                                                     | 21:1516          |
| \CYRHHK . . . . .                                                       | 21:1516          |
| \cyrhhk . . . . .                                                       | 21:1516          |
| \CYRHRDSN . . . . .                                                     | 21:1517          |
| \cyrhrdsn . . . . .                                                     | 21:1516          |
| \CYRI . . . . .                                                         | 21:1517          |
| \cyri . . . . .                                                         | 21:1517          |
| \CYRIE . . . . .                                                        | 21:1517          |
| \cyrie . . . . .                                                        | 21:1517          |
| \CYRII . . . . .                                                        | 21:1517          |
| \cyrii . . . . .                                                        | 21:1517          |
| \CYRISHRT . . . . .                                                     | 21:1517          |
| \cyrishrt . . . . .                                                     | 21:1517          |
| \CYRISHRTDSC . . . . .                                                  | 21:1518          |
| \cyrishrtdsc . . . . .                                                  | 21:1518          |
| \CYRIZH . . . . .                                                       | 21:1518          |
| \cyrizh . . . . .                                                       | 21:1518          |
| \CYRJE . . . . .                                                        | 21:1518          |
| \cyrje . . . . .                                                        | 21:1518          |
| \CYRK . . . . .                                                         | 21:1518          |
| \cyrk . . . . .                                                         | 21:1518          |
| \CYRKBEAK . . . . .                                                     | 21:1519          |
| \cyrkbeak . . . . .                                                     | 21:1519          |
| \CYRKDSC . . . . .                                                      | 21:1519          |
| \cyrkdsc . . . . .                                                      | 21:1519          |
| \CYRKHCRS . . . . .                                                     | 21:1519          |
| \cyrkhcrs . . . . .                                                     | 21:1519          |
| \CYRKHK . . . . .                                                       | 21:1520          |
| \cyrkhk . . . . .                                                       | 21:1519          |
| \CYRKVCRS . . . . .                                                     | 21:1520          |
| \cyrkvcrs . . . . .                                                     | 21:1520          |
| \CYRL . . . . .                                                         | 21:1520          |
| \cyrl . . . . .                                                         | 21:1520          |
| \CYRLDSC . . . . .                                                      | 21:1520          |
| \cyrldsc . . . . .                                                      | 21:1520          |
| \CYRLHK . . . . .                                                       | 21:1521          |
| \cyrlhk . . . . .                                                       | 21:1520          |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

|                          |               |                     |                                     |
|--------------------------|---------------|---------------------|-------------------------------------|
| \CYRLJE . . . . .        | 21:1521       | \CYRSHCH . . . . .  | 21:1527                             |
| \cyrlje . . . . .        | 21:1521       | \cyrshch . . . . .  | 21:1527                             |
| \CYRM . . . . .          | 21:1521       | \CYRSHHA . . . . .  | 21:1527                             |
| \cym . . . . .           | 21:1521       | \cyrshha . . . . .  | 21:1527                             |
| \CYRMDSC . . . . .       | 21:1521       | \CYRT . . . . .     | 21:1528                             |
| \cymdsc . . . . .        | 21:1521       | \cyrt . . . . .     | 21:1528                             |
| \CYRMHK . . . . .        | 21:1521       | \CYRTDSC . . . . .  | 21:1528                             |
| \cymhk . . . . .         | 21:1521       | \cyrtdsc . . . . .  | 21:1528                             |
| \CYRN . . . . .          | 21:1522       | \CYRTETSE . . . . . | 21:1528                             |
| \cyn . . . . .           | 21:1522       | \cyrtetse . . . . . | 21:1528                             |
| \CYRNDSC . . . . .       | 21:1522       | \CYRTSHE . . . . .  | 21:1528                             |
| \cyrndsc . . . . .       | 21:1522       | \cyrtshe . . . . .  | 21:1528                             |
| \CYRNG . . . . .         | 21:1522       | \CYRU . . . . .     | 21:1529                             |
| \cyrng . . . . .         | 21:1522       | \cyru . . . . .     | 21:1529                             |
| \CYRNHK . . . . .        | 21:1522       | \CYRUSHRT . . . . . | 21:1529                             |
| \cyrnhk . . . . .        | 21:1522       | \cyrushrt . . . . . | 21:1529                             |
| \CYRNJE . . . . .        | 21:1523       | \CYRV . . . . .     | 21:1529                             |
| \cyrnje . . . . .        | 21:1522       | \cyrv . . . . .     | 21:1529                             |
| \CYRNLHK . . . . .       | 21:1523       | \CYRW . . . . .     | 21:1529                             |
| \cyrnlhk . . . . .       | 21:1523       | \cyrw . . . . .     | 21:1529                             |
| \CYRO . . . . .          | 21:1523       | \CYRY . . . . .     | 21:1529                             |
| \cyro . . . . .          | 21:1523       | \cyry . . . . .     | 21:1529                             |
| \CYROTLD . . . . .       | 21:1523       | \CYRYA . . . . .    | 21:1530                             |
| \cyrotld . . . . .       | 21:1523       | \cyrya . . . . .    | 21:1530                             |
| \CYRP . . . . .          | 21:1523       | \CYRYAT . . . . .   | 21:1530                             |
| \cyrp . . . . .          | 21:1523       | \cyryat . . . . .   | 21:1530                             |
| \CYRPHK . . . . .        | 21:1524       | \CYRYHCRS . . . . . | 21:1530                             |
| \cyrphk . . . . .        | 21:1524       | \cyryhcrs . . . . . | 21:1530                             |
| \CYRQ . . . . .          | 21:1524       | \CYRYI . . . . .    | 21:1530                             |
| \cyrq . . . . .          | 21:1524       | \cyri . . . . .     | 21:1530                             |
| \CYRR . . . . .          | 21:1524       | \CYRYO . . . . .    | 21:1531                             |
| \cyrr . . . . .          | 21:1524       | \CYRYO . . . . .    | 21:1531                             |
| \CYRDSC . . . . .        | 21:1524       | \cyryo . . . . .    | 21:1530                             |
| \cyrrdsc . . . . .       | 21:1524       | \CYRYU . . . . .    | 21:1531                             |
| \CYRRHK . . . . .        | 21:1525, 1238 | \cyryu . . . . .    | 21:1531                             |
| \cyrrhk . . . . .        | 21:1524, 1238 | \CYRZ . . . . .     | 21:1531                             |
| \CYRRHOOK . . . . .      | 1238          | \cyrz . . . . .     | 21:1531                             |
| \cyrrhook . . . . .      | 1238          | \CYRZDSC . . . . .  | 21:1531                             |
| \CYRRTICK . . . . .      | 21:1525       | \cyrzdsc . . . . .  | 21:1531                             |
| \cyrrtick . . . . .      | 21:1525       | \CYRZH . . . . .    | 21:1531                             |
| \CYRS . . . . .          | 21:1525       | \cyrzh . . . . .    | 21:1531                             |
| \cyrs . . . . .          | 21:1525       | \CYRZHDSC . . . . . | 21:1532                             |
| \CYRSACRS . . . . .      | 21:1525       | \cyrzhdsc . . . . . | 21:1532                             |
| \crysacrs . . . . .      | 21:1525       |                     |                                     |
| \CYRSCHWA . . . . .      | 21:1526       |                     |                                     |
| \cryschwa . . . . .      | 21:1526       | \d . . . . .        | 21:235, 21:395, 21:482, 21:765,     |
| \CYRSDSC . . . . .       | 21:1526       |                     | 21:1249, 21:1463, 21:1464, 21:1465, |
| \crysdesc . . . . .      | 21:1526       |                     | 21:1466, 21:1469, 21:1470, 21:1471, |
| \CYRSEMISFTSN . . . . .  | 21:1526       |                     | 21:1472, 21:1473, 21:1474, 21:1475, |
| \cryssemisftsn . . . . . | 21:1526       |                     | 21:1476, 21:1477, 21:1478, 21:1479, |
| \CYRSFTSN . . . . .      | 21:1527       |                     | 21:1480, 21:1481, 21:1482, 21:1483, |
| \crysftsn . . . . .      | 21:1527       |                     | 21:1484, 21:1485, 21:1486, 21:1487, |
| \CYRSH . . . . .         | 21:1527       |                     | 21:1488, 21:1489, 21:1490, 21:1491, |
| \crysht . . . . .        | 21:1527       |                     | 21:1492, 21:1493, 21:1494, 21:1495, |

**D**

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

21:1496, 21:1497, 21:1498, 21:1499,  
 21:1500, 21:1501, 21:1502, 1222  
`\dag` ..... 21:312, 1218  
`\dagger` ..... 21:312,  
 22:213, 22:219, 22:227, 22:228, 30:382  
`\dashbox` ..... 42:309, 42:807, 42:824  
`\dashv` ..... 30:410  
`\date` ..... 898  
`\date` ..... 44:9, 44:25  
`\day` 01:169, 50:1298, 50:1431, 50:1520, 02:384  
`\dblfigrule` ..... 54:771, 54:2363, 1209  
`\dblfloatpagefraction` .....  
 ..... 45:287, 45:301, 54:2340  
`\dblfloatsep` ..... 54:757,  
 54:769, 54:1644, 54:1770, 54:2347  
`\dbltextfloatsep` ..... 54:224, 54:232,  
 54:773, 54:1643, 54:1769, 54:2347  
`\dbltopfraction` .. 45:284, 45:298, 54:2339  
`\ddag` ..... 21:313, 1218  
`\ddagger` ..... 21:313,  
 22:214, 22:220, 22:227, 22:229, 30:381  
`\ddot` ..... 30:525  
`\ddots` ..... 30:520  
`\deadcycles` 20:335, 20:391, 20:430, 37:33,  
 37:98, 37:174, 37:240, 54:301, 1234  
 debug commands:  
     `\debug_resume`: .....  
         .. 11:102, 11:118, 11:126, 11:134, 224  
     `\debug_suspend`: .....  
         .. 11:98, 11:106, 11:122, 11:130, 224  
`\DebugHooksOff` ..... 08:2798, 08:2906, 206  
`\DebugHooksOn` ..... 08:2798, 08:2905, 304  
`\DebugMarksOff` ..... 48:321, 946  
`\DebugMarksOn` ..... 48:321, 946  
`\DebugShipoutsOff` .. 53:414, 53:446, 1064  
`\DebugShipoutsOn` .. 53:414, 53:445, 1064  
`\DebugSocketsOff` ..... 10:170, 10:191, 335  
`\DebugSocketsOn` ..... 10:170, 10:190, 335  
`\DebugTablesOff` ..... 55:117  
`\DebugTablesOn` ..... 55:117  
 declare commands:  
     `declare_callback_rule` ..... 04:864  
`\declare_callback_rule` ..... 46  
`\DeclareAccent` ..... 1202  
`\DeclareCaseChangeEquivalent` 57:571, 1262  
`\DeclareCommandCopy` .....  
 ..... 06:472, 06:473, 06:475, 1252  
`\DeclareCurrentRelease` .. 33:813, 50:1763  
`\DeclareDefaultHookRule` .....  
 ..... 08:2802, 08:2909, 211  
`\DeclareDocumentCommand` ..... 07:2804  
`\DeclareDocumentEnvironment` ... 07:2840  
`\DeclareEmphSequence` ..... 652

**File Key:** 01=`ltdirchk.dtx`, 02=`ltpplain.dtx`, 03=`ltvers.dtx`, 04=`ltluatex.dtx`, 05=`ltexpl.dtx`,  
 06=`ltdefns.dtx`, 07=`ltcmd.dtx`, 08=`lthooks.dtx`, 09=`ltcmhooks.dtx`, 10=`ltsockets.dtx`,  
 11=`lttemplates.dtx`, 12=`ltalloc.dtx`, 13=`ltcntrl.dtx`, 14=`lterror.dtx`, 15=`ltpar.dtx`,  
 16=`ltpara.dtx`, 17=`ltmeta.dtx`, 18=`ltspace.dtx`, 19=`ltlogos.dtx`, 20=`ltfiles.dtx`,  
 21=`ltoutenc.dtx`, 22=`ltcounts.dtx`, 23=`ltlength.dtx`, 24=`ltfssbas.dtx`, 25=`ltfssaxes.dtx`,  
 26=`ltfsstrc.dtx`, 27=`ltfscmp.dtx`, 28=`ltfssdcl.dtx`, 29=`ltfssini.dtx`, 30=`fontdef.dtx`,  
 31=`preload.dtx`, 32=`ltfntcmd.dtx`, 33=`lttextcomp.dtx`, 34=`ltpageno.dtx`, 35=`ltxref.dtx`,  
 36=`ltproperties.dtx`, 37=`ltmisen.dtx`, 38=`ltmath.dtx`, 39=`ltlists.dtx`, 40=`ltboxes.dtx`,  
 41=`lttab.dtx`, 42=`ltpictur.dtx`, 43=`ltthm.dtx`, 44=`ltsect.dtx`, 45=`ltfloat.dtx`,  
 46=`ltidxglo.dtx`, 47=`ltbibl.dtx`, 48=`ltmarks.dtx`, 49=`ltpage.dtx`, 50=`ltclass.dtx`,  
 51=`ltkeys.dtx`, 52=`ltfilehook.dtx`, 53=`ltshipout.dtx`, 54=`ltoutput.dtx`, 55=`lttagging.dtx`,  
 56=`lthyphen.dtx`, 57=`ltfinal.dtx`

33:1157, 33:1158, 33:1159, 33:1160,  
 33:1161, 33:1162, 33:1163, 33:1164,  
 33:1165, 33:1166, 33:1167, 33:1168,  
 33:1169, 33:1170, 33:1171, 33:1172,  
 33:1173, 33:1174, 33:1175, 33:1176,  
 33:1177, 33:1178, 33:1179, 33:1180,  
 33:1181, 33:1182, 33:1183, 33:1184,  
 33:1185, 33:1186, 33:1187, 33:1188,  
 33:1189, 33:1190, 33:1191, 33:1192,  
 33:1193, 33:1194, 33:1195, 33:1196,  
 33:1197, 33:1198, 33:1199, 33:1200,  
 33:1201, 33:1202, 33:1203, 33:1204,  
 33:1205, 33:1206, 33:1207, 33:1208  
`\DeclareEncodingSubset` ..... 727  
`\DeclareEnvironmentCopy` .....  
 ..... 06:513, 06:514, 06:516, 1264  
`\DeclareErrorFont` .....  
 ..... 24:502, 28:383, 29:735, 30:57  
`\DeclareExpandableDocumentCommand` ..  
 ..... 07:2923  
`\DeclareFixedFont` .....  
 ..... 24:75, 33:1249, 33:1251, 728  
`\DeclareFontEncoding` .....  
 ... 21:377, 21:463, 21:716, 21:738,  
 21:744, 21:830, 21:1038, 24:118,  
 30:133, 30:134, 30:135, 30:136, 1202  
`\DeclareFontEncodingDefaults` .....  
 ..... 24:168, 27:90, 27:91, 30:36  
`\DeclareFontFamily` 24:93, 27:85, 27:86, 539  
`\DeclareFontFamilySubstitution` . 24:462  
`\DeclareFontSeriesChangeRule` .. 25:3,  
 25:4, 25:6, 25:7, 25:8, 25:9, 25:10,  
 25:11, 25:12, 25:13, 25:14, 25:15,  
 25:16, 25:17, 25:18, 25:19, 25:20,  
 25:21, 25:22, 25:23, 25:24, 25:25,  
 25:26, 25:27, 25:28, 25:29, 25:30,  
 25:31, 25:32, 25:33, 25:34, 25:35,  
 25:36, 25:37, 25:38, 25:39, 25:40,  
 25:41, 25:42, 25:43, 25:44, 25:45,  
 25:46, 25:47, 25:48, 25:49, 25:50,  
 25:51, 25:52, 25:53, 25:54, 25:55,  
 25:56, 25:57, 25:58, 25:59, 25:60,  
 25:61, 25:62, 25:63, 25:64, 25:65,  
 25:66, 25:67, 25:68, 25:69, 25:70,  
 25:71, 25:72, 25:73, 25:74, 25:75,  
 25:76, 25:77, 25:78, 25:79, 25:80,  
 25:81, 25:82, 25:83, 25:84, 25:85,  
 25:86, 25:87, 25:88, 25:89, 25:90,  
 25:91, 25:92, 25:93, 25:94, 25:95,  
 25:96, 25:97, 25:98, 25:99, 25:100,  
 25:101, 25:102, 25:103, 25:104,  
 25:105, 25:106, 25:107, 25:108,  
 25:109, 25:110, 25:111, 25:112,  
 25:113, 25:114, 25:115, 25:116, 25:117, 25:118, 25:119, 25:120,  
 25:121, 25:122, 25:123, 25:124,  
 25:125, 25:126, 25:127, 25:128,  
 25:129, 25:130, 25:131, 25:132,  
 25:133, 25:134, 25:135, 25:136,  
 25:137, 25:138, 25:139, 25:140,  
 25:141, 25:142, 25:143, 25:144,  
 25:145, 25:146, 25:147, 25:148,  
 25:149, 25:150, 25:151, 25:152,  
 25:153, 25:154, 25:155, 25:156,  
 25:157, 25:158, 25:159, 25:160,  
 25:161, 25:162, 25:163, 25:164,  
 25:165, 25:166, 25:167, 25:168,  
 25:169, 25:170, 25:171, 25:172,  
 25:173, 25:174, 25:175, 25:176,  
 25:177, 25:178, 25:179, 25:180,  
 25:181, 25:182, 25:183, 25:184,  
 25:185, 25:186, 25:187, 25:188,  
 25:189, 25:190, 25:191, 25:192,  
 25:193, 25:194, 25:195, 25:196,  
 25:197, 25:198, 25:199, 25:200,  
 25:201, 25:202, 25:203, 25:204,  
 25:205, 25:206, 25:207, 25:208,  
 25:209, 25:210, 25:211, 25:212,  
 25:213, 25:214, 25:215, 25:216,  
 25:217, 25:218, 25:219, 25:220,  
 25:221, 25:222, 25:223, 25:224,  
 25:225, 25:226, 25:227, 25:228,  
 25:229, 25:230, 25:231, 25:232,  
 25:233, 25:234, 25:235, 25:236,  
 25:237, 25:238, 25:239, 25:240,  
 25:241, 25:242, 25:243, 25:244,  
 25:245, 25:246, 25:247, 25:248,  
 25:249, 25:250, 25:251, 25:252,  
 25:253, 25:254, 25:255, 25:256,  
 25:257, 25:258, 25:259, 25:260,  
 25:261, 25:262, 25:263, 25:264,  
 25:265, 25:266, 25:267, 25:268,  
 25:269, 25:270, 25:271, 25:272,  
 25:273, 25:274, 25:275, 25:276,  
 25:277, 25:278, 25:279, 25:280,  
 25:281, 25:282, 25:283, 25:284,  
 25:285, 25:286, 25:287, 25:288,  
 25:289, 25:290, 25:291, 25:292,  
 25:293, 25:294, 25:295, 25:296,  
 25:297, 25:298, 25:299, 25:300,  
 25:301, 25:302, 25:303, 25:304,  
 25:305, 25:306, 25:307, 25:308,  
 25:309, 25:310, 25:311, 25:312,  
 25:313, 25:314, 25:315, 25:316,  
 25:317, 25:318, 25:319, 25:320,  
 25:321, 25:322, 25:323, 25:324,  
 25:325, 25:326, 25:327, 25:328,  
 25:329, 25:330, 25:331, 25:332

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

25:333, 25:334, 25:335, 25:336, 27:123, 27:125, 27:132, [28:614](#),  
 25:337, 25:338, 25:339, 25:340, 28:777, 30:158, 30:159, 30:160, 30:161  
 25:341, 25:342, 25:343, 25:344, **\DeclareMathAlphabetCharacter** ... 28:963  
 25:345, 25:346, 25:347, 25:348, **\DeclareMathDelimiter** .....  
 25:349, 25:350, 25:351, 25:352, ..... [28:965](#), 30:262, 30:263,  
 25:353, 25:354, 25:355, 25:356, 30:264, 30:265, 30:266, 30:267,  
 25:357, 25:358, 25:359, 25:360, 30:270, 30:272, 30:273, 30:570,  
 25:361, 25:362, 25:363, 25:364, 30:572, 30:574, 30:576, 30:578,  
 25:365, 25:366, 25:367, 25:368, 30:581, 30:583, 30:585, 30:587,  
 25:369, 25:370, 25:371, 25:372, 30:589, 30:591, 30:593, 30:595,  
 25:373, 25:374, 25:375, 25:376, 30:597, 30:599, 30:601, 30:603,  
 25:377, 25:378, 25:379, 25:380, 30:605, 30:607, 30:609, 30:611,  
 25:381, 25:382, 25:386, 25:388, 551 30:613, 30:615, 30:617, 30:619, [679](#)  
**\DeclareFontSeriesDefault** ..... 635 **\DeclareMathOperator** ..... 1222  
**\DeclareFontSeriesDefault** .....  
 ..... 29:34, [29:35](#), 29:69, 29:71, **\DeclareMathRadical** ..... [28:1100](#), 30:536  
**\DeclareFontShape** ..... 24:18, **\DeclareMathSizes** .....  
 24:471, 24:472, 24:473, 24:474, ..... [24:205](#), 24:211, 24:233,  
 24:475, 24:476, 24:477, 24:478, 30:164, 30:165, 30:166, 30:167,  
 24:479, 24:480, 24:481, 24:482, 30:168, 30:169, 30:170, 30:171,  
 24:483, 24:484, 24:485, 24:486, 30:172, 30:173, 30:174, 30:175, [1220](#)  
 24:487, 24:488, 24:489, 24:490, **\DeclareMathSizes\*** ..... [24:205](#)  
**\DeclareFontShapeChangeRule** ..... **\DeclareMathSymbol** .....  
 ... 25:523, [25:524](#), 25:541, 25:542, ..... [28:901](#), 28:964, 28:981, 30:176,  
 25:543, 25:544, 25:545, 25:546, 30:177, 30:178, 30:179, 30:180,  
 25:547, 25:548, 25:549, 25:550, 30:181, 30:182, 30:183, 30:184,  
 25:551, 25:552, 25:553, 25:554, 30:185, 30:186, 30:187, 30:188,  
 25:555, 25:556, 25:557, 25:558, 30:189, 30:190, 30:191, 30:192,  
 25:559, 25:560, 25:561, 25:562, 30:193, 30:194, 30:195, 30:196,  
 25:563, 25:564, 25:565, 25:566, 30:197, 30:198, 30:199, 30:200,  
 25:567, 25:568, 25:569, 25:570, 30:201, 30:202, 30:203, 30:204,  
 25:571, 25:572, 25:573, 25:574, 30:205, 30:206, 30:207, 30:208,  
 25:575, 25:576, 25:577, 25:578, 30:209, 30:210, 30:211, 30:212,  
 25:579, 25:580, 25:581, 25:582, 30:213, 30:214, 30:215, 30:216,  
 25:583, 25:584, 25:585, 25:586, 30:217, 30:218, 30:219, 30:220,  
 25:587, 25:588, 25:589, 25:590, 30:221, 30:222, 30:223, 30:224,  
 25:591, 25:592, 25:593, 25:594, 30:225, 30:226, 30:227, 30:228,  
 25:595, 25:596, 25:597, 25:601, 25:603 30:229, 30:230, 30:231, 30:232,  
**\DeclareFontSubstitution** .... 21:745, 30:233, 30:234, 30:235, 30:236,  
 21:831, [24:141](#), 30:26, 30:34, 30:37, 30:237, 30:238, 30:239, 30:240,  
 30:38, 30:137, 30:138, 30:139, 30:140 30:241, 30:242, 30:243, 30:244,  
**\DeclareHookRule** ... [08:2800](#), 08:2908, 30:245, 30:246, 30:247, 30:248,  
 37:50, 37:51, 37:115, 37:116, 204 30:249, 30:250, 30:251, 30:252,  
**\DeclareInstance** ..... [11:1156](#), 348 30:253, 30:254, 30:255, 30:256,  
**\DeclareInstanceCopy** ..... [11:1156](#), 348 30:257, 30:258, 30:268, 30:269,  
**\DeclareKeys** ..... [51:184](#), 1027 30:271, 30:275, 30:276, 30:277,  
**\DeclareLowercaseMapping** ... [57:571](#), 1266 30:278, 30:279, 30:280, 30:281,  
**\DeclareMathAccent** ..... [28:788](#), 30:282, 30:283, 30:284, 30:285,  
 30:523, 30:524, 30:525, 30:526, 30:286, 30:287, 30:288, 30:289,  
 30:527, 30:528, 30:529, 30:530, 30:290, 30:291, 30:292, 30:293,  
 30:531, 30:532, 30:533, 30:534, 30:535 30:294, 30:295, 30:296, 30:297,  
**\DeclareMathAlphabet** ..... 27:119, 30:298, 30:299, 30:300, 30:301,  
 30:302, 30:303, 30:304, 30:305, 30:306, 30:307, 30:308, 30:309,  
 30:310, 30:311, 30:312, 30:313,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

30:314, 30:315, 30:316, 30:317, \DeclareRelease . . . 33:811, 33:812, [50:1669](#)  
 30:318, 30:319, 30:320, 30:321, **DeclareRelease commands:**  
 30:322, 30:323, 30:324, 30:325, \DeclareRelease: . . . . . 50:1672  
 30:326, 30:327, 30:328, 30:329, \DeclareRobustCommand . . . . . 14:4,  
 30:330, 30:331, 30:332, 30:334, 14:11, 14:30, 14:57, 18:7, 18:8, 18:9,  
 30:335, 30:336, 30:337, 30:338, 18:10, 18:11, 18:69, 18:93, 18:377,  
 30:339, 30:340, 30:341, 30:348, 18:453, 18:467, 18:512, 18:537, 19:2,  
 30:349, 30:350, 30:351, 30:352, 19:3, 19:13, 20:469, 20:616, 21:150,  
 30:353, 30:354, 30:356, 30:357, 21:158, 21:307, 21:310, 21:311,  
 30:358, 30:359, 30:360, 30:361, 21:312, 21:313, 21:314, 21:316,  
 30:363, 30:364, 30:365, 30:366, 21:318, 21:320, 22:237, 23:32,  
 30:367, 30:368, 30:371, 30:372, 23:33, 23:34, 24:251, 24:279, 24:280,  
 30:373, 30:374, 30:377, 30:378, 24:281, 24:286, 24:298, 24:308,  
 30:379, 30:380, 30:381, 30:382, 24:316, 24:318, 24:336, 24:666,  
 30:383, 30:384, 30:385, 30:386, 24:675, 25:394, 25:398, 25:407,  
 30:387, 30:388, 30:389, 30:390, 25:409, 25:419, 25:526, 25:531,  
 30:391, 30:392, 30:393, 30:394, 25:536, 25:615, 25:618, 25:626,  
 30:395, 30:396, 30:397, 30:398, 25:627, 25:633, 25:713, 26:115,  
 30:399, 30:400, 30:401, 30:402, 26:164, 29:4, 29:7, 29:10, 29:13,  
 30:403, 30:404, 30:405, 30:406, 29:16, 29:19, 29:22, 29:25, 29:28,  
 30:407, 30:408, 30:409, 30:410, 29:254, 29:277, 29:307, 29:328,  
 30:411, 30:412, 30:413, 30:414, 29:352, 29:363, 29:380, 29:383,  
 30:415, 30:416, 30:417, 30:418, 29:405, 29:410, 29:415, 29:448,  
 30:421, 30:422, 30:425, 30:426, 29:453, 29:458, 29:474, 29:477,  
 30:427, 30:428, 30:429, 30:430, 29:480, 29:483, 29:486, 29:500,  
 30:431, 30:432, 30:433, 30:434, 29:549, 29:550, 29:585, 29:591,  
 30:435, 30:436, 30:437, 30:439, 29:613, 29:615, 29:624, 29:626,  
 30:440, 30:441, 30:442, 30:443, 29:633, 29:649, 29:657, 29:675,  
 30:444, 30:445, 30:448, 30:449, 29:691, 29:708, 30:342, 30:343,  
 30:450, 30:452, 30:453, 30:454, 30:344, 30:355, 30:362, 30:419,  
 30:455, 30:456, 30:457, 30:458, 30:420, 30:451, 30:463, 30:467,  
 30:459, 30:460, 30:461, 30:462, 30:470, 30:475, 30:477, 30:479,  
 30:484, 30:486, 30:508, 30:509, 30:482, 30:485, 30:487, 30:488,  
 30:510, 30:560, 30:561, 30:562, 30:490, 30:492, 30:494, 30:496,  
 30:563, 30:621, 30:622, 30:623, 675, 30:498, 30:500, 30:502, 30:504,  
**\DeclareMathVersion . . . 28:396, 29:2, 29:3**  
**\DeclareOldFontCommand . . . 32:125, 32:141**  
**\DeclareOption . . . . . 977**  
**\DeclareOption . . . . .**  
 . . . . . 21:1535, 26:29, 26:37, 26:45, 30:506, 30:512, 30:514, 30:516,  
 26:53, 26:56, 26:60, 33:818, 33:819, 30:519, 30:537, 30:540, 30:543,  
 33:820, 33:821, 33:822, 33:824, 30:547, 30:551, 30:554, 30:557,  
 33:826, 33:828, 33:829, 33:869, 30:564, 30:567, 30:624, 30:625,  
 33:870, 33:871, 33:872, 33:873, 30:626, 30:631, 30:633, 30:635,  
 33:875, 33:876, [50:496](#), 30:1193, 1224 30:637, 32:3, 32:126, 33:4, 33:11,  
**\DeclareOption\* . . . . . 977**  
**\DeclareOption\* . . . . . 50:496**  
**\DeclarePreloadSizes . . . . .**  
 . . . . . 24:185, 27:95, 27:96, 33:610, 33:1133, 35:178, 35:189,  
 31:19, 31:21, 31:22, 31:23, 31:25, 37:269, 37:450, 37:455, 37:460,  
 31:26, 31:27, 31:28, 31:29, 31:30, 37:470, 37:474, 37:478, 37:576,  
 31:34, 31:38, 31:43, 31:45, 31:49, 37:580, 38:3, 38:4, 38:5, 38:6, 38:7,  
 31:50, 31:53, 31:54, 31:57, 31:58, 31:64, 38:8, 38:9, 38:10, 38:11, 38:12,  
**\DeclareProtectedCommand . . . . . 1211**  
 38:13, 38:14, 38:15, 38:16, 38:17, 38:18, 38:19, 38:20, 38:21, 38:22,  
 38:23, 38:24, 38:25, 38:26, 38:27,  
 38:28, 38:29, 38:30, 38:31, 38:32,  
 38:33, 38:34, 38:35, 38:39, 38:41,  
 38:42, 38:43, 38:44, 38:45, 38:46,  
 38:47, 38:48, 38:49, 38:50, 38:51,  
 38:52, 38:81, 38:82, 38:83, 38:84,

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmddhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltospace.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfnctcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisenc.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthypen.dtx, 57=ltfinal.dtx

|                                      |                  |          |                  |                                  |                |             |
|--------------------------------------|------------------|----------|------------------|----------------------------------|----------------|-------------|
| 38:126,                              | 38:167,          | 38:169,  | 38:173,          | \DeclareTextCommand . . . . .    | <u>21:3</u> ,  | 21:58,      |
| 38:218,                              | 38:220,          | 38:222,  | 38:224,          | 21:65,                           | 21:83,         | 21:389,     |
| 38:227,                              | 38:228,          | 38:230,  | 38:237,          | 21:409,                          | 21:410,        | 21:411,     |
| 38:239,                              | 38:240,          | 38:251,  | 38:274,          | 21:415,                          | 21:422,        | 21:424,     |
| 38:276,                              | 38:292,          | 38:303,  | 38:353,          | 21:428,                          | 21:434,        | 21:436,     |
| 38:354,                              | 38:355,          | 38:429,  | 38:463,          | 21:445,                          | 21:475,        | 21:478,     |
| 38:487,                              | 40:7,            | 40:24,   | 40:159,          | 21:485,                          | 21:487,        | 21:490,     |
| 40:195,                              | 40:196,          | 40:212,  | 40:223,          | 21:494,                          | 21:510,        | 21:568,     |
| 40:302,                              | 40:534,          | 40:552,  | 40:560,          | 21:570,                          | 21:730,        | 21:757,     |
| 40:615,                              | 40:616,          | 40:617,  | 40:618,          | 21:762,                          | 21:765,        | 21:798,     |
| 40:619,                              | 40:620,          | 41:139,  | 41:142,          | 21:832,                          | 21:835,        | 21:865,     |
| 41:154,                              | 42:124,          | 42:127,  | 42:130,          | 21:1053,                         | 21:1080,       | 33:376,     |
| 44:7,                                | 44:8,            | 44:9,    | 44:10,           | 33:378,                          | 33:379,        | 33:380,     |
| 44:222,                              | <u>06:233</u> ,  | 45:402,  | 45:423,          | 33:382,                          | 33:383,        | 33:384,     |
| 47:16,                               | 47:28,           | 48:482,  | 48:483,          | 33:625,                          | 33:632,        | 33:639,     |
| 49:34,                               | 49:51,           | 49:59,   | 49:82,           | 33:639,                          | 33:759,        | <u>1218</u> |
| 49:86,                               | 49:93,           | 50:1128, | 50:1129,         | \DeclareTextCommandDefault . . . | <u>21:57</u> , |             |
| 50:1130,                             | 50:1136,         | 50:1137, | 50:1793,         | 21:186,                          | 21:188,        | 21:283,     |
| 52:148,                              | 52:184,          | 53:448,  | 54:87,           | 21:287,                          | 21:288,        | 21:290,     |
| 06:833,                              | 06:834,          | 06:840,  | 06:855,          | 21:296,                          | 21:300,        | 21:301,     |
| \DeclareSizeFunction . . . . .       | <u>26:417</u> ,  | 21:304,  | 21:305,          | 21:306,                          | 21:326,        |             |
| 26:490,                              | 26:491,          | 26:502,  | 26:503,          | 21:355,                          | 33:155,        | 33:157,     |
| 26:507,                              | 26:508,          | 26:514,  | 26:515,          | 33:162,                          | 33:164,        | 33:166,     |
| 26:543,                              | 26:557,          | 26:558,  | 26:565,          | 33:170,                          | 33:172,        | 33:174,     |
| \DeclareSymbolFont . . . . .         | <u>27:136</u> ,  | 33:178,  | 33:180,          | 33:182,                          | 33:184,        |             |
| 28:455,                              | 30:148,          | 30:149,  | 30:150,          | 33:186,                          | 33:188,        | 33:190,     |
| \DeclareSymbolFontAlphabet . . . . . | <u>28:1174</u> , | 33:194,  | 33:195,          | 33:196,                          | 33:197,        |             |
| \DeclareTemplateCode . . . . .       | <u>30:155</u> ,  | 33:198,  | 33:199,          | 33:200,                          | 33:201,        |             |
| 11:994,                              | 11:1035,         | 11:1061, | <u>11:1156</u> , | 33:202,                          | 33:203,        | 33:204,     |
| \DeclareTemplateCopy . . . . .       | <u>30:156</u> ,  | 33:206,  | 33:207,          | 33:208,                          | 33:210,        |             |
| \DeclareTemplateInterface . . .      | <u>30:157</u>    | 33:212,  | 33:214,          | 33:216,                          | 33:218,        |             |
| \DeclareTextAccent . . . . .         | <u>21:64</u> ,   | 33:220,  | 33:222,          | 33:224,                          | 33:226,        |             |
| 21:378,                              | 21:379,          | 21:380,  | 21:381,          | 33:228,                          | 33:230,        | 33:232,     |
| 21:382,                              | 21:383,          | 21:384,  | 21:385,          | 33:236,                          | 33:238,        | 33:240,     |
| 21:386,                              | 21:387,          | 21:388,  | 21:464,          | 33:244,                          | 33:246,        | 33:248,     |
| 21:465,                              | 21:466,          | 21:467,  | 21:468,          | 33:252,                          | 33:254,        | 33:256,     |
| 21:469,                              | 21:470,          | 21:471,  | 21:472,          | 33:260,                          | 33:262,        | 33:264,     |
| 21:473,                              | 21:474,          | 21:741,  | 21:746,          | 33:268,                          | 33:270,        | 33:272,     |
| 21:747,                              | 21:748,          | 21:749,  | 21:750,          | 33:276,                          | 33:278,        | 33:280,     |
| 21:751,                              | 21:752,          | 21:753,  | 21:754,          | 33:284,                          | 33:286,        | 33:288,     |
| 21:755,                              | 21:756,          | 21:838,  | 21:839,          | 33:292,                          | 33:294,        | 33:296,     |
| 21:840,                              | 21:841,          | 21:842,  | 21:843,          | 33:300,                          | 33:302,        | 33:304,     |
| 21:844,                              | 21:845,          | 21:846,  | 21:847,          | 33:309,                          | 33:311,        | 33:313,     |
| 21:848,                              | 21:849,          | 21:850,  | 21:851,          | 33:317,                          | 33:319,        | 33:321,     |
| \DeclareTextAccentDefault . . . . .  | <u>21:852</u>    | 33:325,  | 33:327,          | 33:329,                          | 33:331,        |             |
| ... 21:185,                          | 21:226,          | 21:227,  | 21:228,          | 33:333,                          | 33:335,        | 33:337,     |
| 21:229,                              | 21:230,          | 21:231,  | 21:232,          | 33:341,                          | 33:343,        | 33:345,     |
| 21:233,                              | 21:234,          | 21:235,  | 21:236,          | 33:349,                          | 33:351,        | 33:353,     |
| 21:237,                              | 21:238,          | 21:239,  | 21:279,          | 33:357,                          | 33:359,        | 33:361,     |
| 21:282,                              | 33:687,          | 33:688,  | 33:937,          | 33:365,                          | 33:654,        | 33:662,     |
| 33:938,                              | 33:939,          | 33:940,  | 33:941,          | 33:670,                          | 33:678,        | 33:1003,    |
| 33:942,                              | 33:943,          | 33:944,  | 33:945,          | 33:1006,                         | 33:1008,       | 33:1010,    |
| 33:946,                              | 33:947,          | 33:948,  | 33:949,          | 33:1014,                         | 33:1016,       | 33:1018,    |
|                                      |                  |          |                  | 33:1022,                         | 33:1024,       | 33:1026,    |
|                                      |                  |          |                  | 33:1028,                         | 33:1030,       | 33:1032,    |
|                                      |                  |          |                  | 33:1034,                         | 33:1036,       |             |

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspace.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfnctcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltvphen.dtx, 57=ltfinal.dtx

```

33:1038, 33:1040, 33:1042, 33:1044, \DeclareTextFoo 1202
33:1046, 33:1048, 33:1050, 33:1052, \DeclareTextGlyph 1209
33:1054, 33:1056, 33:1058, 33:1060, \DeclareTextSymbol 21:3, 21:398,
33:1062, 33:1064, 33:1066, 33:1068, 21:399, 21:400, 21:401, 21:402,
33:1070, 33:1072, 33:1074, 33:1076, 21:403, 21:404, 21:405, 21:406,
33:1078, 33:1080, 33:1082, 33:1084, 21:407, 21:408, 21:412, 21:413,
33:1086, 33:1088, 33:1090, 33:1092, 21:416, 21:417, 21:418, 21:419,
33:1094, 33:1096, 33:1098, 33:1100, 21:420, 21:421, 21:526, 21:527,
33:1102, 33:1104, 33:1106, 33:1108, 21:528, 21:529, 21:530, 21:531,
33:1110, 33:1112, 33:1114, 33:1116, 21:532, 21:533, 21:534, 21:535,
33:1118, 33:1120, 33:1123, 1217 21:536, 21:537, 21:538, 21:539,
\DeclareTextComposite . 21:76, 21:452, 21:540, 21:541, 21:543, 21:544,
21:453, 21:587, 21:588, 21:589, 21:545, 21:546, 21:547, 21:548,
21:590, 21:591, 21:592, 21:593, 21:549, 21:550, 21:551, 21:552,
21:594, 21:595, 21:596, 21:597, 21:553, 21:554, 21:555, 21:556,
21:598, 21:599, 21:600, 21:601, 21:557, 21:558, 21:559, 21:560,
21:602, 21:603, 21:604, 21:605, 21:561, 21:562, 21:563, 21:564,
21:606, 21:607, 21:608, 21:609, 21:565, 21:566, 21:567, 21:571,
21:610, 21:611, 21:612, 21:613, 21:572, 21:573, 21:574, 21:575,
21:614, 21:615, 21:616, 21:617, 21:576, 21:577, 21:578, 21:579,
21:618, 21:619, 21:620, 21:621, 21:580, 21:581, 21:582, 21:583,
21:622, 21:623, 21:624, 21:625, 21:584, 21:585, 21:586, 21:717,
21:626, 21:627, 21:628, 21:629, 21:718, 21:719, 21:720, 21:721,
21:630, 21:631, 21:632, 21:633, 21:722, 21:723, 21:724, 21:725,
21:634, 21:635, 21:636, 21:637, 21:726, 21:727, 21:728, 21:729,
21:638, 21:639, 21:640, 21:641, 21:739, 21:740, 21:768, 21:769,
21:642, 21:643, 21:644, 21:645, 21:770, 21:771, 21:772, 21:773,
21:646, 21:647, 21:648, 21:649, 21:774, 21:776, 21:777, 21:778,
21:650, 21:651, 21:652, 21:653, 21:779, 21:780, 21:781, 21:782,
21:654, 21:655, 21:656, 21:657, 21:783, 21:784, 21:785, 21:786,
21:658, 21:659, 21:660, 21:661, 21:787, 21:788, 21:789, 21:790,
21:662, 21:663, 21:664, 21:665, 21:791, 21:792, 21:793, 21:794,
21:666, 21:667, 21:668, 21:669, 21:853, 21:854, 21:855, 21:856,
21:670, 21:671, 21:672, 21:673, 21:857, 21:858, 21:859, 21:860,
21:674, 21:675, 21:676, 21:677, 21:861, 21:862, 21:863, 21:864,
21:678, 21:679, 21:680, 21:681, 21:876, 21:877, 21:878, 21:879,
21:682, 21:683, 21:684, 21:685, 21:880, 21:881, 21:882, 21:883,
21:686, 21:687, 21:688, 21:689, 21:884, 21:885, 21:886, 21:887,
21:690, 21:691, 21:692, 21:693, 21:888, 21:889, 21:890, 21:891,
21:694, 21:695, 21:696, 21:697, 21:892, 21:899, 21:900, 21:901,
21:812, 21:813, 21:814, 21:815, 21:902, 21:903, 21:904, 21:905,
21:816, 21:817, 21:818, 21:819, 21:906, 21:907, 21:908, 21:909,
21:820, 21:821, 21:822, 21:823, 21:910, 21:911, 21:912, 21:913,
21:824, 21:825, 21:826, 21:827, 1226 21:914, 21:915, 21:916, 21:917,
\DeclareTextCompositeCommand . 21:76, 21:918, 21:919, 21:920, 21:921,
21:431, 21:454, 21:455, 21:456, 21:922, 21:923, 21:924, 21:925,
21:457, 21:459, 21:698, 21:699, 21:926, 21:927, 21:928, 21:929,
21:701, 21:704, 21:705, 21:706, 21:930, 21:931, 21:932, 21:933,
21:707, 21:708, 21:709, 21:710, 21:934, 21:935, 21:936, 21:937,
21:711, 21:712, 21:795, 21:1059, 1218 21:938, 21:939, 21:940, 21:941,
\DeclareTextFontCommand 32:1, 21:942, 21:943, 21:944, 21:945,
32:15, 32:16, 32:17, 32:18, 32:19, 21:946, 21:947, 21:948, 21:949,
32:20, 32:21, 32:22, 32:23, 32:24, 21:950, 21:951, 21:952, 21:953,
32:29, 32:30, 32:31, 32:42, 32:140, 1235 21:954, 21:955, 21:956, 21:957,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

21:958, 21:959, 21:960, 21:961, 500  
 21:962, 21:963, 21:964, 21:965,  
 21:966, 21:967, 21:968, 21:969,  
 21:970, 21:971, 21:972, 21:973,  
 21:974, 21:975, 21:976, 21:977,  
 21:978, 21:1079, 33:368, 33:369,  
 33:370, 33:371, 33:372, 33:373,  
 33:374, 33:375, 33:386, 33:387,  
 33:388, 33:389, 33:390, 33:391,  
 33:392, 33:393, 33:394, 33:395,  
 33:589, 33:590, 33:591, 33:592,  
 33:593, 33:594, 33:595, 33:596, 1218  
**\DeclareTextSymbolDefault** . . . . .  
 ... 21:185, 21:240, 21:241, 21:242,  
 21:243, 21:244, 21:245, 21:246,  
 21:247, 21:248, 21:249, 21:250,  
 21:251, 21:252, 21:253, 21:254,  
 21:255, 21:256, 21:257, 21:258,  
 21:259, 21:260, 21:261, 21:262,  
 21:263, 21:264, 21:265, 21:266,  
 21:267, 21:268, 21:269, 21:270,  
 21:271, 21:272, 21:273, 21:274,  
 21:275, 21:276, 21:277, 21:278,  
 21:280, 21:281, 21:291, 33:114,  
 33:116, 33:118, 33:120, 33:121,  
 33:122, 33:123, 33:124, 33:125,  
 33:126, 33:127, 33:128, 33:129,  
 33:130, 33:131, 33:132, 33:133,  
 33:134, 33:135, 33:136, 33:137,  
 33:138, 33:139, 33:140, 33:141,  
 33:142, 33:143, 33:144, 33:145,  
 33:146, 33:147, 33:148, 33:149,  
 33:150, 33:151, 33:152, 33:153,  
 33:154, 33:577, 33:578, 33:579,  
 33:580, 33:581, 33:582, 33:583,  
 33:584, 33:597, 33:598, 33:599,  
 33:600, 33:601, 33:602, 33:603,  
 33:604, 33:623, 33:624, 33:642,  
 33:643, 33:644, 33:645, 33:646,  
 33:647, 33:648, 33:650, 33:950,  
 33:951, 33:952, 33:953, 33:954,  
 33:955, 33:956, 33:957, 33:958,  
 33:959, 33:960, 33:961, 33:962,  
 33:963, 33:964, 33:965, 33:966,  
 33:967, 33:968, 33:969, 33:970,  
 33:971, 33:972, 33:973, 33:974,  
 33:975, 33:976, 33:977, 33:978,  
 33:979, 33:980, 33:981, 33:982,  
 33:983, 33:984, 33:985, 33:986,  
 33:987, 33:988, 33:989, 33:990,  
 33:991, 33:992, 33:993, 33:994,  
 33:995, 33:996, 33:997, 33:998,  
 33:999, 33:1000, 33:1001, 33:1002, 1217  
**\DeclareTitlecaseMapping** . . . 57:571, 1266  
**\DeclareUnicode** . . . . . 500  
**\DeclareUnicodeAccent** . . . . .  
 21:1042, 21:1237, 21:1238, 21:1239,  
 21:1240, 21:1241, 21:1242, 21:1243,  
 21:1244, 21:1245, 21:1246, 21:1247,  
 21:1248, 21:1249, 21:1250, 21:1251, 500  
**\DeclareUnicodeCharacter** . . . 57:384, 1247  
**\DeclareUnicodeCommand** . . . . .  
 ... 21:1080, 21:1081,  
 21:1082, 21:1083, 21:1164, 21:1166,  
 21:1168, 21:1215, 21:1252, 1260  
**\DeclareUnicodeComposite** . . . . .  
 21:1057, 21:1256, 21:1257, 21:1258,  
 21:1259, 21:1260, 21:1261, 21:1262,  
 21:1263, 21:1264, 21:1265, 21:1266,  
 21:1267, 21:1268, 21:1269, 21:1270,  
 21:1271, 21:1272, 21:1273, 21:1274,  
 21:1275, 21:1276, 21:1277, 21:1278,  
 21:1279, 21:1280, 21:1281, 21:1282,  
 21:1283, 21:1284, 21:1285, 21:1286,  
 21:1287, 21:1288, 21:1289, 21:1290,  
 21:1291, 21:1292, 21:1293, 21:1294,  
 21:1295, 21:1296, 21:1297, 21:1298,  
 21:1299, 21:1300, 21:1301, 21:1302,  
 21:1303, 21:1304, 21:1305, 21:1306,  
 21:1307, 21:1308, 21:1309, 21:1310,  
 21:1311, 21:1312, 21:1313, 21:1314,  
 21:1315, 21:1316, 21:1317, 21:1318,  
 21:1319, 21:1320, 21:1321, 21:1322,  
 21:1323, 21:1324, 21:1325, 21:1326,  
 21:1327, 21:1328, 21:1329, 21:1330,  
 21:1331, 21:1332, 21:1333, 21:1334,  
 21:1335, 21:1336, 21:1337, 21:1338,  
 21:1339, 21:1340, 21:1341, 21:1342,  
 21:1343, 21:1344, 21:1345, 21:1346,  
 21:1347, 21:1348, 21:1349, 21:1350,  
 21:1351, 21:1352, 21:1353, 21:1354,  
 21:1355, 21:1356, 21:1357, 21:1358,  
 21:1359, 21:1360, 21:1361, 21:1362,  
 21:1363, 21:1364, 21:1365, 21:1366,  
 21:1367, 21:1368, 21:1369, 21:1370,  
 21:1371, 21:1372, 21:1373, 21:1374,  
 21:1375, 21:1376, 21:1377, 21:1378,  
 21:1379, 21:1380, 21:1381, 21:1382,  
 21:1383, 21:1384, 21:1385, 21:1386,  
 21:1387, 21:1388, 21:1389, 21:1390,  
 21:1391, 21:1392, 21:1393, 21:1394,  
 21:1395, 21:1396, 21:1397, 21:1398,  
 21:1399, 21:1400, 21:1401, 21:1402,  
 21:1403, 21:1404, 21:1405, 21:1406,  
 21:1407, 21:1408, 21:1409, 21:1410,  
 21:1411, 21:1412, 21:1413, 21:1414,  
 21:1415, 21:1416, 21:1417, 21:1418,  
 21:1419, 21:1420, 21:1421, 21:1422

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefsns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspace.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthypphen.dtx, 57=ltfinal.dtx

```

\dimenzero 04:226 \DocInput 26:8, 30:5, 31:5, 56:4
\dimeval 77 \docclearpage 1239
\dimeval 05:158, 05:174, 1260 \document 442
\dimexpr 21:503, 21:519, document (env.) 37:8
 21:869, 21:872, 21:1219, 21:1222, 42:13 \document 16:143, 20:9, 37:258, 47:64, 47:81, 1240
\directlua 04:2, 04:14, 04:29, \documentclass 04:16, 26:2, 30:2,
 04:213, 04:231, 21:1011, 04:256, 31:2, 50:674, 50:681, 50:740, 50:743,
 04:261, 04:265, 01:12, 01:15, 06:34, 50:975, 50:1070, 50:1202, 56:2, 975
 01:20, 01:23, 50:1254, 50:1385, \DocumentMetadata 17:6, 17:11, 17:15, 17:18, 17:61, 418
 06:754, 06:839, 01:28, 57:98, 57:100, \documentstyle 50:679, 50:1202
 57:108, 57:113, 57:115, 02:65, \doesglyphexist 33:1218,
 02:81, 02:105, 02:256, 02:353, 02:363 33:1244, 33:1262, 33:1263, 33:1264,
disable commands: 33:1265, 33:1266, 33:1267, 33:1268,
 disable_callback 04:975 33:1271, 33:1272, 33:1273, 33:1274,
\disable_callback 46 33:1277, 33:1278, 33:1279, 33:1280,
\DisableGenericHook 08:2738, 08:2744, 08:2892, 306 33:1283, 33:1284, 33:1287, 33:1288,
\DisableHook 08:2843, 195 33:1291, 33:1292, 33:1293, 33:1294,
\DiscardShipoutBox 53:82, 53:411, 53:444, 53:500, 1063 33:1295, 33:1296, 33:1297, 33:1298,
\discretionary 38:251, 06:841, 06:856, 06:866 33:1301, 33:1302, 33:1305, 33:1306,
\displayindent 02:403 33:1307, 33:1308, 33:1309, 33:1310,
\displaylines 38:203 33:1311, 33:1312, 33:1313, 33:1314,
displaymath (env.) 38:345 33:1315, 33:1316, 33:1317, 33:1318,
\displaymath 38:347 33:1319, 33:1320, 33:1321, 33:1322,
\displaystyle 30:539, 33:1323, 33:1324, 33:1325, 33:1326,
 30:542, 30:546, 30:548, 38:62, 33:1327, 33:1328, 33:1329, 33:1330,
 38:210, 38:372, 38:375, 38:428, 33:1331, 33:1332, 33:1333, 33:1334,
 38:470, 38:482, 38:510, 38:535, 38:538 33:1335, 33:1336, 33:1337, 33:1338,
\displaywidowpenalty 02:327 33:1339, 33:1340, 33:1341, 33:1342,
\displaywidth 38:210, 38:371, 38:457, 38:513, 02:402 33:1343, 33:1344, 33:1345, 33:1346,
\div 30:390 33:1347, 33:1348, 33:1349, 33:1350,
\DJ 21:529, 21:1138, 57:650, 1213 33:1351, 33:1352, 33:1353, 33:1354,
\dj 21:539, 21:1139, 57:650, 1213 33:1355, 33:1356, 33:1357, 33:1358,
\do 01:110, 13:3, 13:7, 13:16, 13:26, 33:1359, 33:1360, 33:1361, 33:1362,
 20:66, 20:69, 20:72, 20:135, 20:138, 33:1363, 33:1364, 33:1365, 33:1366,
 20:190, 20:193, 20:233, 20:305, 33:1367, 33:1368, 33:1369, 33:1370,
 20:367, 20:414, 20:587, 20:604, 33:1373, 33:1377, 33:1378, 33:1379,
 20:755, 20:761, 32:90, 37:541, 33:1380, 33:1381, 33:1382, 33:1383,
 37:562, 37:722, 37:732, 37:738, 33:1384, 33:1385, 33:1386, 33:1387,
 37:744, 40:78, 40:97, 06:69, 41:244, 33:1388, 33:1389, 33:1390, 33:1391,
 41:269, 42:104, 42:116, 42:123, 33:1392, 33:1393, 33:1394, 33:1395,
 42:203, 42:337, 42:339, 42:362, 33:1396, 33:1397, 33:1398, 33:1399,
 42:365, 42:394, 42:396, 42:417, 33:1400, 33:1401, 33:1402, 33:1403,
 42:422, 42:477, 42:505, 42:534, 33:1404, 33:1405, 33:1406, 33:1407, 729
 42:730, 42:786, 45:65, 45:134, 47:36, \dollar 1214
 47:65, 47:82, 50:232, 50:249, 50:543, \dospecials 01:110, 37:541, 37:562,
 50:562, 50:579, 50:597, 50:602, 37:722, 37:732, 50:1301, 50:1434,
 50:616, 50:621, 50:657, 50:667, 50:1523, 02:13, 07:1953, 01:58, 1231
 50:1182, 50:1219, 50:1301, 50:1354, \dot 30:532
 50:1434, 50:1523, 50:1808, 02:13, \doteq 30:476
 02:14, 07:1952, 01:58, 01:59, 446 \dotfill 06:881, 06:902, 02:546
 \dots 21:320, 21:322, 1221 \doublehyphendemerits 02:334

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\doublerulesep 41:311, 41:338, 41:362
\Downarrow 30:593
\downarrow 30:587
\downbracefill 30:545, 30:564
\dump 57:714
\endarray 41:171
\endcenter 37:445
\endcsname 200
\enddisplaymath 38:348
\enddocument 37:8, 37:77,
 37:78, 37:141, 37:143, 53:352, 1227
\endenumerate 39:264
\endeqnarray 38:380, 38:427
\endequation 38:351
\endfilecontents 50:1206
\endflushleft 37:495
\endflushright 37:497
\endgraf 16:140,
 16:178, 40:138, 40:144, 02:445, 1206
\endgroup 1194
\EndIncludeInRelease
 08:2136, 08:2143, 08:2167,
 08:2183, 08:2190, 08:2195, 08:2217,
 02:579, 08:2237, 08:2267, 08:2285,
 08:2290, 08:2300, 08:2304, 08:2307,
 08:2334, 08:2349, 08:2360, 08:2378,
 08:2390, 08:2402, 08:2412, 08:2442,
 08:2461, 08:2524, 02:613, 08:2528,
 02:621, 08:2610, 08:2613, 08:2727,
 08:2733, 08:2740, 08:2745, 08:2752,
 08:2756, 08:2763, 08:2767, 08:2790,
 08:2795, 02:646, 09:31, 09:45,
 02:686, 02:691, 09:228, 09:295,
 02:701, 02:706, 09:420, 09:439,
 09:491, 09:495, 09:542, 09:569,
 09:572, 09:584, 03:71, 14:157,
 14:161, 14:204, 14:210, 16:44,
 16:75, 17:43, 17:54, 18:21, 18:31,
 18:65, 18:78, 18:86, 18:91, 18:104,
 18:110, 18:152, 18:166, 18:178,
 18:189, 18:206, 18:218, 18:250,
 18:266, 18:283, 18:319, 18:353,
 18:375, 18:412, 18:445, 18:481,
 18:485, 18:494, 18:498, 18:506,
 18:510, 18:520, 18:526, 18:540,
 18:547, 20:83, 20:140, 20:195,
 20:256, 20:277, 20:289, 20:352,
 20:356, 20:435, 20:473, 20:496,
 20:531, 20:552, 20:571, 20:578,
 20:597, 20:614, 20:627, 20:631,
 20:649, 20:660, 20:670, 20:709,
 20:736, 21:103, 21:123, 21:168,
 21:175, 21:330, 21:352, 21:367,
 21:375, 04:234, 04:257, 04:280,
 04:284, 22:28, 22:33, 22:57, 22:75,
 22:86, 22:91, 22:116, 22:121, 22:146,
 22:173, 22:182, 22:224, 22:230,
 22:250, 22:253, 23:10, 23:14, 23:25,
 23:30, 24:52, 24:73, 24:231, 24:248,
\endverbvisiblespace 37:343

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

24:294, 24:304, 24:314, 24:366,  
 24:375, 24:455, 24:460, 24:495,  
 24:500, 24:517, 24:535, 24:574,  
 24:607, 24:724, 24:736, 25:384,  
 25:390, 25:403, 25:415, 25:422,  
 25:505, 25:520, 25:599, 25:611,  
 25:622, 25:629, 25:636, 25:709,  
 25:727, 25:734, 25:738, 25:741,  
 26:157, 26:160, 26:176, 26:549,  
 26:555, 27:21, 27:143, 28:77, 28:105,  
 28:179, 28:209, 28:239, 28:271,  
 28:306, 28:349, 28:410, 28:416,  
 28:421, 28:504, 28:511, 28:556,  
 28:563, 28:837, 28:879, 28:892,  
 28:899, 28:1087, 28:1095, 29:67,  
 29:100, 29:122, 29:227, 29:248,  
 29:300, 29:347, 29:376, 29:387,  
 29:434, 29:468, 29:494, 29:530,  
 29:539, 29:576, 29:588, 29:595,  
 29:629, 29:635, 29:670, 29:686,  
 29:703, 29:718, 30:88, 30:98, 30:116,  
 30:125, 30:640, 30:647, 32:33, 32:40,  
 05:10, 05:18, 05:27, 35:28, 35:45,  
 35:61, 35:75, 35:82, 35:110, 35:114,  
 35:138, 35:147, 35:158, 35:168,  
 35:180, 35:191, 35:202, 37:75,  
 37:139, 37:178, 05:110, 37:186,  
 37:192, 37:225, 37:238, 37:266,  
 37:277, 37:308, 37:333, 05:131,  
 37:341, 37:359, 37:374, 37:382,  
 37:387, 37:401, 37:410, 37:420,  
 37:427, 37:437, 37:442, 37:466,  
 37:482, 37:491, 05:147, 37:510,  
 37:516, 37:546, 37:567, 37:589,  
 05:156, 37:599, 37:616, 37:628,  
 37:632, 37:640, 37:644, 37:665,  
 37:678, 37:693, 37:705, 37:727,  
 37:735, 38:86, 38:95, 05:168, 38:117,  
 38:124, 38:143, 38:148, 38:156,  
 38:160, 05:176, 38:175, 38:183,  
 38:232, 38:249, 38:279, 38:287,  
 38:316, 38:343, 05:191, 38:407,  
 38:416, 05:197, 38:448, 38:453,  
 38:473, 38:485, 38:494, 38:503,  
 05:203, 05:208, 06:12, 39:134,  
 06:18, 39:139, 39:154, 39:162,  
 40:13, 40:22, 40:57, 40:65, 40:90,  
 40:107, 40:121, 40:132, 40:141,  
 40:146, 40:152, 40:163, 40:170,  
 40:227, 40:234, 40:260, 40:271,  
 40:307, 40:315, 40:354, 40:375,  
 40:399, 40:417, 40:494, 40:510,  
 40:527, 40:536, 40:541, 40:564,  
 40:571, 40:605, 40:612, 06:60,  
 06:64, 41:64, 41:69, 41:156, 41:164,  
 41:226, 41:231, 42:15, 42:19, 42:38,  
 42:47, 42:68, 42:76, 42:88, 42:96,  
 42:111, 42:121, 42:150, 42:155,  
 42:171, 42:182, 42:249, 42:263,  
 42:368, 42:425, 42:462, 42:468,  
 42:499, 42:529, 42:554, 42:570,  
 42:581, 42:594, 42:602, 42:623,  
 42:640, 42:650, 42:654, 42:744,  
 42:799, 42:818, 42:835, 43:33,  
 43:39, 43:56, 43:63, 44:19, 44:29,  
 44:167, 44:173, 44:178, 44:192,  
 44:198, 44:224, 44:246, 45:104,  
 06:224, 45:172, 06:229, 45:231,  
 45:246, 45:293, 45:306, 45:351,  
 45:368, 45:411, 45:417, 45:426,  
 45:430, 45:439, 45:445, 45:449,  
 45:481, 45:498, 45:516, 45:558,  
 45:564, 47:24, 47:32, 47:76, 47:92,  
 01:290, 06:324, 49:47, 49:68, 49:75,  
 50:21, 50:26, 06:352, 50:49, 50:61,  
 50:84, 50:93, 50:99, 50:111, 50:142,  
 50:149, 50:174, 50:181, 50:199,  
 50:210, 50:244, 50:261, 50:272,  
 50:281, 06:380, 50:312, 06:385,  
 50:342, 50:359, 50:371, 50:392,  
 50:405, 50:418, 50:428, 50:464,  
 50:480, 50:489, 50:523, 01:300,  
 50:533, 06:404, 50:573, 50:589,  
 50:611, 50:626, 50:640, 50:647,  
 50:662, 06:418, 50:671, 50:705,  
 50:712, 50:729, 50:736, 50:789,  
 50:796, 50:834, 50:861, 50:889,  
 50:1045, 50:1102, 50:1132, 50:1139,  
 50:1157, 50:1167, 06:459, 06:469,  
 50:1342, 50:1473, 50:1563, 06:501,  
 06:510, 06:537, 06:543, 52:14, 52:23,  
 52:88, 52:140, 52:179, 52:191,  
 06:557, 52:200, 06:562, 52:245,  
 52:256, 52:263, 52:298, 52:321,  
 52:342, 52:356, 52:361, 52:384,  
 52:404, 52:429, 52:474, 52:494,  
 52:501, 52:531, 52:536, 01:25,  
 53:429, 53:466, 53:483, 53:487,  
 06:663, 54:53, 54:62, 54:180, 06:678,  
 54:198, 54:367, 54:372, 54:420,  
 54:466, 54:654, 54:712, 54:815,  
 06:729, 54:834, 06:733, 54:897,  
 54:915, 54:957, 54:978, 06:765,  
 54:1220, 06:774, 54:1389, 54:1471,  
 54:1565, 54:1687, 54:1814, 54:1924,  
 54:1932, 06:824, 54:1966, 54:1995,  
 06:831, 54:2215, 54:2233, 54:2280,  
 06:853, 54:2324, 06:864, 06:867,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

06:895, 57:15, 57:19, 57:29, 57:33,  
 57:51, 57:69, 57:79, 57:86, 57:94,  
 06:916, 57:141, 57:146, 57:198,  
 57:222, 57:292, 57:297, 57:338,  
 57:344, 57:440, 57:487, 07:237,  
 07:260, 02:87, 01:34, 02:101, 02:118,  
 02:123, 02:133, 02:137, 02:147,  
 02:150, 07:1142, 07:1145, 07:1160,  
 07:1173, 07:1176, 02:167, 07:1306,  
 07:1309, 02:181, 07:1338, 07:1341,  
 07:1350, 07:1358, 07:1361, 02:185,  
 07:1521, 07:1525, 02:219, 02:224,  
 02:232, 02:240, 07:1993, 07:2004,  
 02:288, 02:300, 07:2892, 07:2922,  
 07:2992, 07:2999, 02:359, 02:367,  
 08:85, 08:101, 08:129, 08:145,  
 08:157, 08:169, 08:185, 08:198,  
 08:218, 08:228, 08:248, 08:263,  
 08:281, 08:287, 08:307, 08:325,  
 08:329, 08:578, 08:629, 08:646,  
 08:650, 08:673, 08:694, 08:728,  
 08:786, 08:816, 08:848, 08:875,  
 08:878, 08:898, 08:901, 08:914,  
 08:932, 08:937, 08:940, 08:949,  
 08:954, 08:957, 08:994, 02:471,  
 08:1031, 08:1081, 08:1088, 08:1104,  
 08:1111, 08:1131, 08:1135, 02:488,  
 08:1187, 08:1191, 08:1207, 08:1211,  
 08:1239, 08:1242, 08:1265, 08:1269,  
 08:1286, 08:1291, 08:1298, 08:1302,  
 08:1338, 08:1362, 08:1445, 08:1472,  
 08:1506, 08:1545, 08:1634, 08:1717,  
 08:1948, 08:2052, 08:2057, 08:2062, 28  
`\EndIncludeRelease` ..... 03:78  
`\endinput` ..... 1207  
`\enditemize` ..... 39:275  
`\endline` ..... 38:188, 02:445  
`\endlinechar` ..... 20:684,  
 20:718, 01:188, 06:37, 06:39, 06:44,  
 50:433, 50:434, 50:435, 57:329,  
 02:380, 01:76, 01:77, 01:78, 1239  
`\endlist` ..... 39:98, 39:264, 39:275  
`\endlrbbox` ..... 40:194  
`\endmath` ..... 38:346  
`\endminipage` ..... 40:460  
`\EndModuleRelease` ..... 08:2914,  
 09:608, 10:193, 03:111, 03:112,  
 03:152, 11:1199, 16:183, 17:63,  
 33:809, 36:280, 48:493, 55:249, 07:3012  
`\endpicture` ..... 42:49  
`\endscname` ..... 1193  
`\endsloppypar` ..... 49:92  
`\endtabbing` ..... 41:84  
`\endtabular` ..... 41:171  
`\endtabular*` ..... 41:171  
`\endtrivlist` .....  
 ... 37:445, 37:495, 37:497, 37:570,  
 38:515, 39:100, 39:101, 41:85, 43:66  
`\endverbatim` ..... 37:569, 37:587  
`\enlargethispage` ..... 54:1859, 953  
`\enlargethispage*` ..... 54:1859  
`\enskip` ..... 18:549  
`\enspace` ..... 18:537, 18:545  
`\ensuremath` ..... 22:226, 38:429,  
 45:409, 45:416, 45:437, 45:444, 1230  
`enumerate` (env.) ..... 39:255  
`\enumerate` ..... 39:255  
environments:  
`array` ..... 41:168  
`center` ..... 37:444  
`displaymath` ..... 38:345  
`document` ..... 37:8  
`enumerate` ..... 39:255  
`eqnarray` ..... 38:357, 38:516  
`eqnarray*` ..... 38:425  
`equation` ..... 38:349, 38:504  
`filecontents` ..... 50:1206, 975  
`flushleft` ..... 37:494  
`flushright` ..... 37:496  
`itemize` ..... 39:266  
`list` ..... 39:34  
`lrbox` ..... 818  
`math` ..... 38:345  
`minipage` ..... 819  
`picture` ..... 42:21  
`sloppypar` ..... 49:91  
`tabbing` ..... 41:71  
`tabular` ..... 41:174  
`thebibliography` ..... 937  
`trivlist` ..... 39:89  
`verbatim` ..... 37:569  
`verbatim*` ..... 37:584  
`\epsilon` ..... 30:279  
`eqnarray` (env.) ..... 38:357, 38:516  
`\eqnarray` ..... 38:362, 38:426  
`eqnarray*` (env.) ..... 38:425  
`\eqno` ..... 38:351, 38:439, 1217  
`equation` (env.) ..... 38:349, 38:504  
`\equation` ..... 38:350  
`\equiv` ..... 30:455  
`\errhelp` ..... 03:31, 14:39,  
 14:66, 01:201, 56:12, 57:307, 57:703  
`\errmessage` ..... 03:32, 10:72,  
 14:47, 14:72, 04:63, 01:7, 24:546,  
 24:581, 26:425, 26:525, 27:65,  
 01:206, 05:79, 05:94, 56:16, 57:63,  
 57:309, 01:42, 02:164, 02:178, 02:304

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\ERROR . . . . . 07:1638, 07:1648, 07:1657,  
   07:2029, 07:2042, 07:2062, 07:2081,  
   07:2157, 07:2184, 08:1718, 08:1719, 169  
 \errorcontextlines . . . . . 02:569,  
   02:588, 02:604, 02:619, 02:633,  
   02:657, 02:674, 14:212, 02:389, 1201  
 \ERRORmissingcells . . . . . 55:145  
 \errorstopmode . . . . . 57:713, 02:554, 1226  
 \ERRORusetaggingsocket . . . . . 55:26, 55:36  
 \escapchar . . . . . 455  
 \escapechar . . . . . 20:454, 04:212,  
   24:392, 24:616, 26:229, 28:58, 28:86,  
   28:158, 28:189, 28:219, 28:250,  
   28:372, 06:126, 06:169, 06:173,  
   06:181, 06:287, 06:288, 06:312,  
   06:449, 52:280, 52:303, 52:326,  
   52:347, 06:572, 06:590, 02:377, 102  
 \eta . . . . . 30:281  
 \etacatcode . . . . . 04:1026  
 \TeXversion . . . . . 01:41  
 \evensidemargin . . . . . 54:73, 54:617, 54:676  
 \everycr . . . . . 38:205,  
   38:208, 38:371, 38:532, 02:534, 1198  
 \everydisplay . . . . . 24:350, 24:351, 24:360, 24:371  
 \everyeof . . . . . 57:329  
 \everyjob . . . . . 03:37, 04:216, 04:262, 04:263,  
   28:392, 53:29, 53:30, 57:112, 57:331,  
   57:416, 57:673, 57:674, 57:676, 38  
 \everymath . . . . . 24:349, 24:351, 24:364, 24:373  
 \everypar . . . . . 399  
 \everypar . . . . . 16:35, 16:41, 16:72, 16:77,  
   16:120, 16:130, 16:143, 16:181,  
   20:43, 20:113, 20:171, 24:669,  
   24:683, 24:730, 37:241, 37:543,  
   37:565, 39:130, 39:131, 39:133,  
   39:137, 39:138, 39:204, 39:221,  
   40:389, 40:410, 41:81, 44:48, 44:96,  
   44:107, 44:127, 44:136, 45:187,  
   54:167, 54:194, 54:1155, 54:1321, 416  
 \EveryShipout . . . . . 1066  
 \ExecuteOptions . . . . . 26:57, 26:70, 33:877, 50:650  
 \exhyphenpenalty . . . . . 02:322, 02:507  
 \exists . . . . . 30:331  
 \exp . . . . . 38:31  
 exp commands:  
   \exp:w . . . . . 07:2399, 08:757, 08:811, 246  
   \exp\_after:wN . . . . . 08:2102,  
     08:2231, 08:2415, 08:2418, 08:2482,  
     08:2944, 09:170, 09:245, 09:299,  
     09:300, 09:353, 09:471, 11:285,  
     11:740, 11:873, 11:880, 16:41, 16:72,  
     28:319, 28:322, 49:41, 51:46, 52:73,  
     52:236, 52:237, 07:322, 07:338,  
     07:392, 07:398, 07:412, 07:1134,  
     07:1154, 07:1158, 07:1169, 07:1171,  
     07:1220, 07:1269, 07:1299, 07:1330,  
     07:1337, 07:1413, 07:1547, 07:1607,  
     07:1631, 07:1647, 07:1665, 07:1671,  
     07:1675, 07:1706, 07:1740, 07:1747,  
     07:1760, 07:1766, 07:1772, 07:1778,  
     07:1784, 07:1790, 07:1796, 07:1802,  
     07:1833, 07:1961, 07:2402, 07:2403,  
     07:2449, 07:2450, 07:2451, 07:2452,  
     07:2453, 07:2454, 07:2455, 07:2486,  
     08:56, 08:61, 08:380, 08:381,  
     08:426, 08:756, 08:810, 08:1040,  
     08:1042, 08:1183, 08:1348, 08:1521,  
     08:1731, 08:1840, 08:1873, 08:1958, 246  
 \exp\_args:Nc . . . . . 08:2172, 09:29,  
   09:42, 09:74, 09:132, 48:22, 07:263,  
   07:1065, 07:1076, 07:1376, 07:1417,  
   07:1436, 07:1540, 08:1204, 08:1217  
 \exp\_args:Ncc . . . . . 07:138, 07:180  
 \exp\_args:Ncv . . . . . 08:569  
 \exp\_args:Ne . . . . . 08:2320, 08:2321,  
   09:505, 11:845, 51:110, 51:141,  
   52:46, 52:97, 07:2475, 08:714, 08:716  
 \exp\_args:Nee . . . . . 52:129  
 \exp\_args:Nf . . . . . 09:161,  
   09:236, 52:32, 52:34, 52:394, 07:417,  
   07:2161, 07:2188, 07:2249, 07:2477  
 \exp\_args:NNc . . . . . 09:146, 09:151  
 \exp\_args:NNe . . . . . 07:1346, 08:1623  
 \exp\_args:Nne . . . . . 08:1850  
 \exp\_args:NNNo . . . . . 07:398, 07:1875  
 \exp\_args:Nnnv . . . . . 08:257  
 \exp\_args:NNo . . . . . 08:2945, 09:169,  
   09:244, 08:1064, 08:1121, 08:1712  
 \exp\_args:NNV . . . . .  
   . . . . . 09:188, 09:263, 08:1596, 08:1683  
 \exp\_args:NNx . . . . . 07:1435, 08:1502, 08:1540  
 \exp\_args:No . . . . .  
   . . . . . 08:2120, 08:2934, 08:2945,  
   09:169, 09:188, 09:244, 09:263,  
   36:41, 36:46, 07:90, 07:319, 07:409,  
   07:428, 07:437, 07:525, 07:909,  
   07:1305, 07:2034, 07:2045, 07:2086,  
   07:2167, 07:2194, 07:2417, 07:2443,  
   08:1064, 08:1121, 08:1182, 08:2061  
 \exp\_args:Nof . . . . . 07:2037  
 \exp\_args:Noo . . . . . 07:1278  
 \exp\_args:Nv . . . . . 09:189, 09:264,  
   51:192, 52:53, 07:256, 07:361, 07:1502  
 \exp\_args:Nv . . . . .  
   . . . . . 08:2100, 08:1249, 08:1990, 08:1998  
 \exp\_args:Nx . . . . . 36:10, 53:29, 07:2416,  
   07:2419, 07:2442, 08:432, 08:474, 08:545  
 \exp\_args\_generate:n . . . . . 57:593

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphyphen.dtx, 57=ltfinal.dtx

```

\exp_end: 07:2402, 08:757, 08:811
\exp_last_unbraced:Ne
..... 08:2936, 07:2488,
08:1066, 08:1099, 08:1123, 08:1216
\exp_last_unbraced:Nf
..... 09:110, 08:1229, 08:1897
\exp_last_unbraced:NNf 08:1157
\exp_last_unbraced:NNNNo
.... 09:116, 09:474, 08:383, 08:2055
\exp_last_unbraced:NnNo 07:2261
\exp_last_unbraced:NNo 09:364
\exp_not:N .. 09:108, 09:110, 09:111,
09:112, 09:184, 09:199, 09:206,
09:207, 09:214, 09:219, 09:259,
09:274, 09:281, 09:282, 09:286,
09:290, 09:307, 09:352, 09:358,
09:376, 09:385, 09:459, 09:460,
09:480, 09:485, 09:508, 09:535,
09:536, 09:563, 09:564, 11:277,
11:278, 11:279, 11:280, 11:290,
11:296, 11:299, 11:301, 11:302,
11:306, 11:308, 11:311, 11:316,
11:460, 11:461, 11:476, 11:478,
11:494, 11:667, 11:669, 11:772,
11:780, 11:820, 11:835, 11:860,
11:867, 16:39, 16:40, 16:70, 16:71,
16:79, 16:80, 36:72, 48:101, 48:104,
48:106, 48:111, 48:114, 51:192,
51:193, 51:194, 53:31, 53:95, 53:125,
57:600, 07:120, 07:135, 07:136,
07:164, 07:165, 07:197, 07:199,
07:200, 07:206, 07:208, 07:267,
07:279, 07:296, 07:297, 07:298,
07:300, 07:368, 07:918, 07:928,
07:998, 07:1011, 07:1035, 07:1053,
07:1060, 07:1098, 07:1106, 07:1107,
07:1139, 07:1185, 07:1186, 07:1193,
07:1194, 07:1195, 07:1199, 07:1202,
07:1203, 07:1205, 07:1224, 07:1227,
07:1228, 07:1229, 07:1237, 07:1284,
07:1303, 07:1317, 07:1324, 07:1325,
07:1426, 07:1427, 07:1807, 07:1808,
07:1916, 07:1918, 07:1919, 07:1985,
07:1988, 07:2001, 07:2478, 07:2481,
07:2482, 07:2483, 07:2484, 07:2486,
08:391, 08:1151, 08:1155, 08:1174,
08:1177, 08:1178, 08:1180, 08:1224,
08:1486, 08:1488, 08:1624, 08:1626, 315
\exp_not:n
.. 08:2934, 09:171, 09:184, 09:187,
09:189, 09:224, 09:246, 09:259,
09:262, 09:264, 09:292, 09:356,
09:386, 09:388, 09:389, 09:391,
09:392, 11:282, 11:463, 11:588,
11:668, 11:773, 11:774, 11:781,
11:819, 11:821, 11:822, 11:835,
11:838, 11:839, 11:860, 11:861,
11:867, 11:868, 36:238, 48:236,
48:237, 48:238, 48:263, 48:328,
48:329, 48:330, 48:332, 48:333,
48:334, 48:336, 48:337, 48:338,
48:340, 48:341, 48:342, 48:344,
48:345, 48:346, 48:348, 48:349,
48:350, 51:176, 52:211, 52:225,
53:30, 07:159, 07:163, 07:167,
07:170, 07:174, 07:191, 07:198,
07:210, 07:280, 07:299, 07:368,
07:371, 07:372, 07:625, 07:759,
07:865, 07:876, 07:878, 07:879,
07:947, 07:999, 07:1012, 07:1023,
07:1035, 07:1052, 07:1138, 07:1140,
07:1193, 07:1206, 07:1238, 07:1273,
07:1531, 07:1533, 07:1546, 07:1616,
07:1617, 07:1660, 07:1720, 07:1917,
07:2052, 07:2053, 07:2093, 07:2094,
07:2276, 07:2285, 07:2302, 07:2316,
07:2328, 07:2478, 08:114, 08:241,
08:548, 08:1048, 08:1072, 08:1171,
08:1175, 08:1181, 08:1182, 08:1630, 124
\exp_stop_f: 07:1458,
07:2039, 08:1250, 08:1398, 08:1406
expandable commands:
\expandable_grab_(type):w 147
\expandafter 1221
expandafter commands:
\expandafter: 32:88, 41:269
\ExpandArgs 78
\ExpandArgs 05:177,
05:193, 05:196, 06:534, 06:535, 57:597
\expanded 50:902, 57:111, 236
\ExplLoaderFileDate 05:201, 05:207
\ExplSyntaxOff 08:2951,
09:381, 09:609, 10:194, 11:1200,
16:184, 17:42, 17:64, 20:529, 20:749,
20:799, 28:304, 28:347, 36:281,
05:146, 05:166, 05:189, 48:494,
49:45, 50:515, 51:257, 52:12, 52:86,
52:138, 52:243, 52:254, 52:402,
52:427, 52:472, 52:529, 52:584,
53:427, 53:474, 53:514, 55:62,
55:239, 57:570, 57:648, 07:3013, 981
\ExplSyntaxOn
.... 09:3, 09:381, 10:2, 11:6, 16:3,
17:2, 17:22, 20:518, 20:743, 20:796,
28:277, 28:310, 36:2, 05:135, 05:161,
05:180, 48:3, 49:22, 50:513, 51:3,
52:7, 52:29, 52:92, 52:207, 52:251,
52:390, 52:410, 52:435, 52:508,
```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

52:577, 53:5, 53:472, 53:509, 55:2,  
 55:99, 07:8, 57:560, 57:571, 08:3, 304  
`\extracolsep` ..... 41:167  
`\extrafloats` ..... 02:152, 02:189, 02:274

**F**

`\fam` 04:22, 04:26, 04:38, 24:15, 02:98, 02:375  
`\family` ..... 1194  
`\familydefault` .....  
 .. 21:1574, 28:389, 29:444, 29:659,  
 29:677, 29:693, 29:710, 30:127, 642  
`\fbox` ..... 818  
`\fbox` ..... 40:212, 40:225, 40:232, 1211  
`\fboxrule` 40:210, 40:257, 40:269, 40:274,  
 40:280, 40:282, 40:289, 40:290, 57:151  
`\fboxsep` 40:210, 40:216, 40:249, 40:256,  
 40:268, 40:275, 40:285, 40:287, 57:150  
`\fi` ..... 1241  
`fi` commands:  
`\fi:` ..... 08:2103, 08:2207,  
 08:2215, 08:2232, 08:2254, 08:2282,  
 08:2419, 08:2424, 08:2440, 08:2487,  
 08:2522, 08:2537, 09:342, 09:344,  
 09:345, 09:349, 09:354, 09:361,  
 16:26, 16:33, 16:65, 16:103, 52:241,  
 07:876, 07:879, 07:946, 07:1303,  
 07:1460, 07:2399, 07:2403, 08:769,  
 08:1043, 08:1200, 08:1223, 08:1263,  
 08:1402, 08:1410, 08:1416, 08:1734, 319  
`\filbreak` ..... 02:511  
`file` commands:  
`\g_file_curr_name_str` ..... 08:2781  
`\file_full_name:n` . 20:525, 52:35, 1045  
`\file_mdfive_hash:n` ..... 20:797  
`\file_parse_full_name_apply:nN` ..  
 .... 52:32, 52:47, 52:90, 52:93, 52:95  
`\l_file_search_path_seq` ..... 1037  
`\file_size:n` ..... 20:798  
`file` internal commands:  
`\_file_parse_full_name_area:nw` ..  
 .... 52:102, 52:105, 52:109  
`\_file_parse_full_name_auxi:nN` ..  
 .... 52:97, 52:100  
`\_file_parse_full_name_base:nw` ..  
 .... 52:108, 52:112, 52:124  
`\_file_parse_full_name_tidy:nnnN` ..  
 .... 52:119, 52:120, 52:122, 52:127  
`file/.../after` ..... 1036  
`file/.../before` ..... 1036  
`file/after` ..... 1036  
`file/before` ..... 1036  
`filecontents (env.)` ..... 50:1206, 975  
`\filecontents` ..... 50:1206

`filehook` internal commands:  
`\_filehook_clear_replacement_-`  
`flag:` ..... 52:417, 52:420, 52:520  
`\_filehook_drop_extension:N` ..  
 ..... 52:44, 52:49, 52:522  
`\_filehook_drop_extension_-`  
`aux:nnn` ..... 52:54, 52:57  
`\_filehook_file_name_compose:nnn`  
 ..... 52:218, 52:406, 52:414, 52:415, 52:425  
`\_filehook_file_parse_full_-`  
`name:nN` 52:28, 52:30, 52:30, 52:53,  
 52:238, 52:394, 52:412, 52:414, 1042  
`\_filehook_file_pop:` .....  
 ..... 52:59, 52:70, 52:526  
`\_filehook_file_pop_assign:nnnn`  
 ..... 52:59, 52:73, 52:79, 52:528  
`\_filehook_file_push:` .....  
 ..... 52:59, 52:62, 52:524  
`\_filehook_file_replaced` .... 52:417  
`\_filehook_file_subst_begin:nnn`  
 ..... 52:412, 52:422, 52:422, 1052  
`\_filehook_file_subst_cycle_-`  
`error:NN` ..... 52:457,  
 52:462, 52:462, 52:467, 52:469, 1054  
`\_filehook_file_subst_loop:NN` ..  
 . 52:431, 52:444, 52:452, 52:461, 1053  
`\_filehook_file_subst_tortoise_-`  
`hare:nn` ..... 52:424,  
 52:431, 52:434, 52:436, 52:459, 1054  
`\_filehook_full_name:nn` ..  
 ..... 52:30, 52:34, 52:38  
`\_filehook_if_file_replaced:TF` ..  
 ..... 52:417, 52:418, 52:518, 1053  
`\_filehook_if_no_extension:nTF` ..  
 ..... 52:44, 52:44, 52:510  
`\g_filehook_input_file_seq` ....  
 ..... 52:59, 1043  
`\l_filehook_internal_t1` .... 52:59  
`\_filehook_log_file_record:n` ..  
 ..... 52:547, 52:547, 52:561, 52:563  
`\g_filehook_nesting_level_int` ..  
 52:544, 52:549, 52:552, 52:553, 52:559  
`\_filehook_normalize_file_-`  
`name:w` ..... 52:406, 52:413, 52:516  
`\_filehook_resolve_file_subst:w` ..  
 ..... 52:406, 52:409, 52:411, 52:514  
`\_filehook_set_curr_file:nNN` ..  
 ..... 52:392, 52:512  
`\_filehook_set_curr_file_-`  
`assign:nnnnN` ..... 52:395, 52:397  
`\_filehook_subst_add:nn` ..  
 . 52:206, 52:208, 52:208, 52:252, 1047  
`\_filehook_subst_empty_name_-`  
`chk:NN` ..... 52:208, 52:236, 52:240

**File Key:** 01=ltidirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__filehook_subst_file_normalize:Nn 32:85, 33:6, 33:612, 33:626, 33:633,
 42:126, 42:129, 42:678, 02:539, 02:544
__filehook_subst_remove:n \fontencoding .. 21:867, 04:259, 04:260,
 04:282, 04:283, 21:1573, 24:251,
 24:286, 24:298, 24:308, 28:388,
 29:658, 29:676, 29:692, 30:16, 30:24,
 30:75, 30:82, 30:91, 30:93, 33:36, 1212
\filename@parse 6, 1 \fontfamily .. 24:279, 29:6, 29:9, 29:12,
 29:145, 29:156, 29:482, 29:485,
 29:488, 29:738, 30:66, 30:77, 30:84,
 30:95, 33:28, 33:30, 33:32, 33:34,
 33:85, 33:87, 33:89, 33:91, 33:913, 1196
\filesize 05:72, 05:118 \FontFamilyToCheck
 33:1620, 33:1621, 33:1634
\fill 18:530 \fontname 21:1004, 24:89
\finalhyphendemerits \fontseries .. 24:279,
 25:393, 25:394, 25:405, 25:407,
 25:417, 25:419, 29:15, 29:18, 29:258,
 29:259, 29:260, 29:261, 29:281,
 29:282, 29:283, 29:284, 29:319,
 29:320, 29:321, 29:322, 29:339,
 29:340, 29:341, 29:342, 29:355,
 29:356, 29:357, 29:358, 29:366,
 29:367, 29:368, 29:369, 29:382,
 29:385, 29:476, 29:479, 29:739, 1251
\floatingpenalty \fontseriesforce 25:398, 25:409, 25:420, 560
\floatpagefraction 45:278, 54:2336
\floatsep 54:730, 54:748, \fontshape .. 21:449, 21:809,
 54:755, 54:2142, 54:2192, 54:2341
 24:279, 25:528, 25:533, 25:538,
 25:614, 25:615, 25:624, 25:626,
 25:631, 25:633, 25:684, 25:688,
 25:691, 25:694, 25:697, 25:700,
 25:703, 25:706, 25:713, 29:21, 29:24,
 29:27, 29:30, 29:740, 33:636, 1250
\fmtname 03:1, 03:41, 03:43, 03:46, 03:48, \fontshapeforce
 03:51, 03:60, 50:749, 50:753, 1199
 25:618, 25:627, 25:634, 25:714
\fmtversion 03:1, \fontsize .. 19:6, 21:302,
 03:19, 03:41, 03:43, 03:46, 03:48,
 03:51, 03:60, 03:105, 03:118, 03:166,
 14:2, 04:276, 21:1538, 29:225,
 41:1, 42:1, 50:169, 50:177, 50:766,
 50:769, 54:4, 57:657, 57:683, 1199
 21:328, 21:360, 21:868, 21:1218,
 21:1254, 24:79, 24:318, 29:641,
 29:741, 33:104, 33:672, 33:930,
 45:409, 45:416, 45:437, 45:444, 1212
\fnsymbol 512 \fontsubfuzz .. 26:441,
\fnsymbol 22:189 \FOO 81
\font 21:297, 21:298, \foo 313
 21:299, 21:439, 21:446, 21:799,
 21:806, 21:866, 21:1003, 21:1004,
 21:1060, 21:1165, 21:1167, 21:1169,
 21:1216, 24:81, 24:87, 24:89, 26:84,
 29:553, 29:586, 29:592, 29:618, 31:8,
 31:9, 31:10, 32:85, 33:6, 33:612,
 33:626, 33:633, 37:564, 06:842,
 06:845, 06:857, 06:860, 02:539, 02:544
\fontdimen 21:297, 21:298, \footins .. 45:390, 45:465,
 21:299, 21:439, 21:446, 21:799,
 21:806, 29:553, 29:586, 29:592
 45:484, 45:502, 54:316, 54:317,
 54:318, 54:319, 54:377, 54:424,
 54:486, 54:494, 54:498, 54:521, 1225
\fontencoding 25:156, 25:157, \footnote .. 45:452, 1195
\fontfamily .. 24:279, 29:6, 29:9, 29:12, \footnotemark .. 44:10, 45:518, 1195
\fontseries .. 24:279, 25:393, 25:405, 25:407, \footnoterule .. 40:466, 45:394, 54:497, 1225
\fontshape .. 21:449, 25:528, 25:533, 25:538, \footnotesep .. 40:490,
 25:614, 25:615, 25:624, 25:626, 40:507, 40:524, 45:451, 45:468,
 25:631, 25:633, 25:684, 25:688, 45:477, 45:487, 45:495, 45:505, 45:513
\fontsize .. 19:6, 21:302, 21:328, 21:360, 21:868, 21:1218, 21:1254, 24:79, 24:318, 29:641, 29:741, 33:104, 33:672, 33:930, 45:409, 45:416, 45:437, 45:444, 1212
\fontshapeforce .. 25:618, 25:627, 25:634, 25:714
\fontsubfuzz .. 26:441, 26:475, 37:53, 37:118, 37:157, 1208
\FOO 81
\foo 313
\footins .. 45:390, 45:465, 45:484, 45:502, 54:316, 54:317, 54:318, 54:319, 54:377, 54:424, 54:486, 54:494, 54:498, 54:521, 1225
\footnote .. 45:452, 1195
\footnotemark .. 44:10, 45:518, 1195
\footnoterule .. 40:466, 45:394, 54:497, 1225
\footnotesep .. 40:490, 40:507, 40:524, 45:451, 45:468, 45:477, 45:487, 45:495, 45:505, 45:513

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltyphephen.dtx, 57=ltfinal.dtx

\footnotesize ..... 40:482,  
   40:500, 40:517, 45:466, 45:485, 45:503  
 \footnotetext ..... 44:12, 45:535, 1195  
 \footref ..... 45:547, 1256  
 \footskip ..... 54:77, 54:641, 54:700  
 \forall ..... 30:330  
 fp commands:  
   \fp\_eval:n ..... 11:368, 05:162  
     \l\_tmpa\_fp ..... 346  
 \fpval ..... 77  
 \fpeval ..... 05:158, 05:170, 05:172, 77  
 \frac ..... 38:354, 1201  
 \frame ..... 40:196, 40:297  
 \framebox ..... 818  
 \framebox ..... 40:219, 1216  
 \frenchspacing .....  
   ... 20:46, 20:116, 20:174, 37:569,  
   37:586, 37:737, 06:882, 06:903, 02:431  
 \frown ..... 30:458  
 \fussy ..... 49:93  
 \futurelet ..... 18:457, 18:465, 32:83,  
   05:49, 38:256, 41:359, 06:782, 06:796

**G**

\g\_@@\_\meta\_{hook}\_code\_prop ..... 225  
 \g\_@@\_\meta\_{hook}\_declared\_tl ..... 225  
 \g\_@@\_\meta\_{hook}\_parameter\_tl ..... 225  
 \g\_@@\_\meta\_{hook}\_reversed\_tl ..... 225  
 \Gamma ..... 30:304  
 \gamma ..... 30:277  
 \gcd ..... 38:33  
 \gdef ..... 982  
 gdef commands:  
   \gdef\_ ..... 38:269  
 \ge ..... 30:424, 30:426  
 \Generic\* ..... 1219  
 \GenericError ..... 14:18,  
   14:85, 14:111, 14:164, 26:62, 1215  
 \GenericInfo ... 03:106, 03:108, 03:119,  
   03:122, 03:127, 03:162, 03:163,  
   03:168, 03:172, 03:176, 14:4, 03:197,  
   14:104, 14:130, 14:182, 26:31,  
   26:34, 26:39, 26:75, 05:63, 50:66,  
   50:74, 50:104, 50:107, 50:1734, 1215  
 \GenericWarning ..... 14:11, 14:94,  
   14:120, 14:141, 14:149, 14:173,  
   14:195, 26:42, 26:47, 26:50, 26:78, 1215  
 \geq ..... 30:422, 30:424  
 \getanddefinefonts ..... 1193  
 \GetFileInfo ..... 30:3  
 \gets ..... 30:446, 30:448  
 \gg ..... 30:440  
 \glossary ..... 1249  
 \global ..... 1233

\globaldefs ..... 24:618, 26:231, 28:60, 28:89,  
   28:160, 28:191, 28:221, 28:253, 02:372  
 \glossary ..... 934  
 \glossary .....  
   44:182, 46:23, 46:35, 48:253, 49:25,  
   49:36, 49:53, 49:61, 54:625, 54:684, 943  
 \glossaryentry ..... 46:32  
 \glyphmissingdetails ..... 33:1219, 33:1244, 728  
 \goodbreak ..... 06:883, 06:904, 02:511  
 \grave ..... 30:524  
 \group ..... 1193  
 group commands:  
   \group\_align\_safe\_begin/end: ... 164  
   \group\_align\_safe\_begin: .....  
     ... 07:286, 07:298, 07:403, 122  
   \group\_align\_safe\_end: .....  
     ... 07:320, 07:426, 07:1863, 123  
 \group\_begin: .....  
   09:201, 09:276, 16:17, 16:23, 16:50,  
   16:56, 28:293, 28:332, 48:66, 48:220,  
   51:4, 52:210, 52:224, 53:92, 07:1855,  
   07:2231, 07:2236, 07:2408, 07:2412,  
   07:2440, 08:387, 08:1150, 08:1163  
 \c\_group\_begin\_token ..... 07:682,  
   07:693, 07:1861, 07:1942, 07:2520  
 \group\_end: ..... 09:205,  
   09:280, 16:19, 16:28, 16:52, 16:60,  
   28:296, 28:335, 48:71, 48:240, 51:21,  
   52:220, 52:232, 53:94, 07:1876,  
   07:2240, 07:2245, 07:2422, 07:2423,  
   07:2425, 07:2427, 07:2437, 07:2444,  
   07:2445, 08:390, 08:1154, 08:1173  
 \c\_group\_end\_token ... 07:685, 07:1931  
 \group\_insert\_after:N .....  
   28:318, 28:319, 28:322, 28:344, 28:345  
 \guillemetleft ..... 21:540, 21:773, 21:1104, 1247  
 \guillemetright .....  
   ... 21:541, 21:774, 21:1120, 1247  
 \guillemotleft ... 21:543, 21:776, 21:1106  
 \guillemotright ... 21:544, 21:777, 21:1122  
 \guilsinglleft ..... 21:545, 21:1183  
 \guilsinglright ..... 21:546, 21:1184

**H**

\H ..... 14:24, 21:230, 21:385, 21:469,  
   21:601, 21:609, 21:628, 21:636,  
   21:753, 21:1246, 21:1385, 21:1386,  
   21:1413, 21:1414, 33:177, 33:201  
 \halign ..... 38:127,  
   38:210, 38:371, 38:532, 02:534, 1193  
 \hang ..... 1224  
 \hangafter ..... 02:374  
 \hangindent ..... 44:139, 02:405  
 \hat ..... 30:530

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\hbadness 24:672, 24:679, 24:715, 24:734,  
53:238, 53:240, 53:290, 53:292, 02:318  
\hbar ..... 30:342  
\hbox ..... 965  
hbox commands:  
  \hbox:n ..... 53:382  
  \hbox\_set:Nn .....  
    53:170, 53:217, 53:243, 53:268, 53:297  
  \hbox\_set\_to\_wd:Nnn .....  
    53:241, 53:293, 53:329  
  \hbox\_unpack:N .....  
    48:389, 48:420, 53:251, 53:295  
  \hbox\_unpack\_drop:N ..... 53:180  
\hbox\_upto\_u ..... 1209  
\headheight ..... 54:75, 54:630, 54:689  
\headsep ..... 54:76, 54:639, 54:698  
\heartsuit ..... 30:340  
\height . 21:872, 21:1222, 40:31, 40:34, 1200  
\hfil ..... 803  
\hfill ..... 1196  
\hfuzz 24:680, 49:89, 49:90, 49:96, 49:97,  
53:237, 53:239, 53:289, 53:291, 02:390  
\hglue ..... 02:501  
\hideoutput ..... 02:692  
\hideskip ..... 02:309, 02:525  
\hidewidth ..... 21:327,  
21:329, 21:358, 21:362, 21:390,  
21:391, 21:394, 21:397, 21:476,  
21:477, 21:481, 21:484, 21:486,  
21:489, 21:501, 21:506, 21:522,  
21:760, 21:761, 21:764, 21:767,  
21:834, 21:837, 21:1253, 21:1255, 02:525  
\hline ..... 41:358, 41:361, 1197  
\hoffset ..... 02:406  
\holdinginserts ..... 02:339  
\hom ..... 38:29  
hook commands:  
  \hook\_activate\_generic:n .....  
    08:2737, 08:2823,  
    08:2842, 08:2855, 08:288, 08:288,  
    08:290, 08:308, 08:326, 08:328, 234  
  \hook\_debug\_off: .....  
    08:2799, 08:7, 08:13, 209  
  \hook\_debug\_on: 08:2798, 08:7, 08:8, 209  
  \hook\_disable:n ..... 08:2811, 08:2811  
  \hook\_disable\_generic:n .....  
    08:2739, 08:2816, 08:2848, 08:264,  
    08:264, 08:266, 08:282, 08:285, 207  
  \hook\_gclear\_next\_code:n .....  
    08:2184, 08:2184, 08:2769, 208  
  \hook\_gput\_code:nnn .....  
    .. 08:2749, 08:486, 08:486, 08:488,  
    08:579, 08:581, 08:681, 08:702, 243  
  \hook\_gput\_code\_with\_args:nnn ...  
    .. 08:2751, 08:486, 08:494, 08:628, 208  
  \hook\_gput\_next\_code:nn .....  
    08:2122, 08:2122, 08:2124, 08:2137,  
    08:2139, 08:2760, 08:690, 08:708, 208  
  \hook\_gput\_next\_code\_with\_-  
    args:nn 08:2130, 08:2142, 08:2762, 208  
  \hook\_gremove\_code:nn .. 08:2771,  
    08:958, 08:958, 08:960, 08:995, 208  
  \hook\_gset\_rule:nnnn ... 08:2801,  
    08:2803, 08:2806, 08:1303, 08:1303, 209  
  \hook\_if\_empty:n .....  
    .. 08:2426, 08:2428, 08:2443, 08:2445  
  \hook\_if\_empty:nTF .....  
    08:2426, 08:2807, 08:2808, 08:2809,  
    53:63, 53:164, 08:1862, 08:1968, 200  
  \hook\_if\_empty\_p:n 08:2426, 53:67,  
    53:113, 53:377, 08:1917, 08:2017, 209  
  \hook\_log:n .....  
    .. 08:2797, 08:1818, 08:1818, 209  
  \hook\_new:n ..... 08:2714,  
    48:259, 53:153, 53:154, 53:155,  
    53:156, 53:157, 53:158, 53:159,  
    08:67, 08:69, 08:88, 08:223, 08:861, 206  
  \hook\_new\_pair:nn ..... 08:2718,  
    16:6, 16:7, 08:199, 08:201, 08:221, 206  
  \hook\_new\_pair\_with\_args:nn ... 207  
  \hook\_new\_pair\_with\_args:nnn ...  
    .. 08:2726, 08:199,  
    08:199, 08:203, 08:219, 08:226, 1265  
  \hook\_new\_reversed:n ... 08:2716,  
    08:170, 08:172, 08:188, 08:224, 206  
  \hook\_new\_reversed\_with\_args:nn ..  
    .. 08:2724, 08:170,  
    08:170, 08:174, 08:186, 08:197, 207  
  \hook\_new\_with\_args:nn .. 08:2722,  
    08:67, 08:67, 08:71, 08:86, 08:100, 207  
  \g\_hook\_patch\_action\_list\_tl ...  
    .. 09:6, 09:91, 09:124  
  \hook\_provide:n .... 08:2811, 08:2818  
  \hook\_provide\_pair:nn 08:2811, 08:2832  
  \hook\_provide\_reversed:n .....  
    .. 08:2811, 08:2825  
  \hook\_show:n .....  
    .. 08:2796, 08:1818, 08:1823, 209  
  \hook\_use:n .....  
    08:2196, 08:2196, 08:2198, 08:2218,  
    08:2220, 08:2238, 08:2240, 08:2295,  
    08:2786, 16:22, 16:31, 16:55, 16:63,  
    16:98, 16:105, 48:222, 53:62, 53:65,  
    53:71, 53:75, 53:123, 08:1465, 236  
  \hook\_use:nnw ..... 08:2236,  
    08:2266, 08:2268, 08:2268, 08:2270,  
    08:2286, 08:2288, 08:2296, 08:2788, 208

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

```

\hook_use_once:n 08:2335, 08:2337, 08:2352, 08:2787, 207
\hook_use_once:nnw 08:2335, 08:2335, 08:2343, 08:2350, 08:2358, 08:2789, 208
hook internal commands:
\c_hook 08:1295, 08:1299
\g_hook_??_code_prop 08:1292
\c_hook_??_parameter_tl 08:1292
\g_hook_??_reversed_tl 08:1292
\g_hook_(hook)_code_prop 289
\g_hook_(hook)_labels_clist 229
\g_hook_(hook)_reversed_tl 225
_hook_activate_generic:n .. 08:288
_hook_activate_generic:nn 08:2840, 08:291, 08:292, 08:310
_hook_activate_generic_pair:nn .. 08:2811, 08:2837, 08:2841, 08:2869
_hook_activate_generic_- reversed:n 08:2811, 08:2830, 08:2839, 08:2842, 08:2862
\g_hook_all_seq 08:2942, 08:28, 08:108, 08:137, 08:1426, 08:1453
_hook_apply_-rule_->:nnn .. 08:1793
_hook_apply_-rule_-<:nnn .. 08:1793
_hook_apply_-rule_<:nnn .. 08:1793
_hook_apply_-rule_>:nnn .. 08:1793
_hook_apply_-rule_xE:nnn .. 08:1793
_hook_apply_-rule_xW:nnn .. 08:1793
_hook_apply_label_pair:nnn 08:1573, 08:1574, 08:1655, 08:1656, 08:1720, 08:1720, 280
_hook_apply_rule:nnn 08:1730, 08:1736, 08:1736, 272
_hook_apply_rule:nnnN 272
_hook_apply_rule_->:nnn .. 08:1771
_hook_apply_rule_-<:nnn .. 08:1771
_hook_apply_rule_<:nnn .. 08:1743
_hook_apply_rule_>:nnn .. 08:1743
_hook_apply_rule_xE:nnn .. 08:1757
_hook_apply_rule_xW:nnn .. 08:1757
_hook_braced_cs_parameter:n 08:2937, 08:1067, 08:1124, 08:1149, 08:1160, 08:1212, 08:1212, 08:1214, 08:1240
_hook_braced_hidden_loop:w 08:1212, 08:1216, 08:1219, 08:1225
_hook_braced_parameter:n 09:221, 09:505, 08:1243, 08:1243, 08:1245, 08:1266, 08:1268, 08:1487, 08:1489, 08:1624, 08:1626
_hook_braced_real_loop:w .. 08:1243
_hook_chk_args_allowed:nn 08:2160, 08:502, 08:630, 08:630, 08:632, 08:647, 08:649, 282
_hook_clear_next:n 08:2164, 08:2180, 08:2185, 08:2186, 08:2186, 08:2188, 08:2191, 08:2193, 279
_hook_clist_gput:Nn 08:1495, 08:1497, 08:1529, 08:1533, 08:1597, 08:1684, 08:1718, 08:1719
_hook_cmd_begindocument_code: .. 09:46, 09:54, 09:59, 09:62, 09:606, 327
_hook_cmd_if_scansable:Nn .. 09:450
_hook_cmd_if_scansable:NnTF 09:408, 09:427, 09:450, 324
_hook_cmd_patch_xparse:Nnn 09:128, 09:149, 09:149
_hook_cmd_try_patch:nn 09:52, 09:63, 09:63
_hook_code_gset:nn .. 08:2397, 08:118, 08:1089, 08:1089, 08:1091, 08:1103, 08:1105, 08:1107, 08:1484
_hook_code_gset_aux:nnn 08:1052, 08:1089, 08:1092, 08:1094, 08:1096, 08:1097, 08:1110
_hook_code_gset_auxi:nnnn 08:1032, 08:1053, 08:1078, 08:1080, 08:1087, 08:1118, 253
_hook_cs_end:w 08:1212, 08:1231, 08:1232, 08:1233, 08:1238
_hook_cs_gput_right:nnn 08:2165, 08:2368, 08:540, 08:1032, 08:1032, 08:1034, 08:1082, 08:1084
_hook_cs_gput_right_fast:nnn 08:1032, 08:1040, 08:1046, 08:1085
_hook_cs_gput_right_slow:nnn 08:1032, 08:1042, 08:1049, 08:1086
_hook_cs_if_empty:N 08:1192, 08:1194, 08:1208, 08:1210
_hook_cs_if_empty:NTF 08:2161, 08:2163, 08:2434, 08:2435, 08:1192, 08:1887, 08:1894
_hook_cs_if_empty_p:N 08:1192
_hook_cs_parameter_count:N 08:1212, 08:1217, 08:1227
_hook_cs_parameter_count:w 08:1212, 08:1229, 08:1236, 08:1237
\l_hook_cur_hook_tl 08:29, 08:1559, 08:1641, 08:1777, 08:1788, 273
_hook_curr_name_pop: 08:2777, 08:2883, 08:417, 08:449, 238
_hook_curr_name_push:n 08:2775, 08:2780, 08:417, 08:431, 298
_hook_curr_name_push_aux:n 08:417, 08:432, 08:433
_hook_curname_or_default: 08:2625, 08:2673,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=ltphypen.dtx, 57=ltfinal.dtx

```

08:333, 08:343, 08:347, 08:363,
08:364, 08:364, 08:537, 08:611, 235
__hook_debug:n
. 09:19, 09:28, 09:39, 09:48, 09:49,
09:65, 09:69, 08:7, 08:7, 08:20,
08:531, 08:604, 08:1425, 08:1434,
08:1452, 08:1455, 08:1477, 08:1501,
08:1511, 08:1539, 08:1578, 08:1598,
08:1660, 08:1685, 08:1745, 08:1752,
08:1759, 08:1767, 08:1773, 08:1784, 222
\g_hook_debug_bool
..... 08:6, 08:10, 08:15, 08:21
__hook_debug_gset:
..... 08:7, 08:11, 08:16, 08:18
__hook_debug_label_data:N
..... 08:1578, 08:1620,
08:1661, 08:1709, 08:1806, 08:1806
__hook_debug_print_rules:n
..... 08:2106, 08:2106
__hook_declare_DEPRECATED_-
generic>NNn . 08:756, 08:777, 08:810
__hook_declare_DEPRECATED_-
generic>NNw . 08:772, 08:778, 08:779
__hook_def_cmd:w
..... 09:13, 09:14, 09:192,
09:198, 09:267, 09:273, 09:299, 317
\g_hook_delayed_patches_prop ...
..... 09:17, 09:51, 09:57, 09:58
__hook_DEPRECATED_generic_-
warn:n
08:755,
08:762, 08:809, 08:987, 08:1023,
08:1314, 08:1345, 08:1838, 08:1956, 246
__hook_DEPRECATED_generic_-
warn:Nn
08:762
__hook_DEPRECATED_generic_-
warn:Nw
08:762
__hook_DEPRECATED_generic_-
warn:w
08:763, 08:764
__hook_DEPRECATED_warn:nn
..... 08:2813, 08:2820,
08:2827, 08:2834, 08:2845, 08:2852,
08:2859, 08:2866, 08:2871, 08:2871
__hook_disable:n
09:209, 09:501, 08:264, 08:267, 08:268
__hook_do_DEPRECATED_generic:Nn
.. 08:772, 08:772, 08:988, 08:1024,
08:1315, 08:1346, 08:1839, 08:1957
__hook_do_DEPRECATED_generic:Nw
..... 08:772, 08:773, 08:774
__hook_double_hashes:n
..... 09:188, 09:263,
09:325, 09:325, 09:363, 08:549,
08:1064, 08:1073, 08:1121, 08:1178, 257
__hook_double_hashes:w
..... 09:325, 09:326,
09:327, 09:359, 09:363, 09:366, 319
__hook_double_hashes_group:n ...
..... 09:325, 09:333, 09:362
__hook_double_hashes_output:N ..
..... 09:325, 09:330, 09:338, 319
__hook_double_hashes_space:w ...
..... 09:325, 09:334, 09:365
__hook_double_hashes_stop:w ...
..... 09:325, 09:341, 09:361
\c_hook_empty_tl
08:35, 08:1284
__hook_end_document_label_-
check: 08:417, 08:456, 08:457, 08:464
__hook_exp_not:n
... 09:171, 09:172, 09:176, 09:187,
09:246, 09:247, 09:251, 09:262, 314
__hook_exp_not>NNN
09:13,
09:13, 09:193, 09:199, 09:219,
09:224, 09:268, 09:274, 09:290, 09:292
__hook_file_hook_normalize:n ...
..... 08:2321, 08:716,
08:899, 08:899, 08:902, 08:904, 249
\l_hook_front_tl
08:1548,
08:1589, 08:1592, 08:1595, 08:1597,
08:1598, 08:1599, 08:1613, 08:1614,
08:1674, 08:1678, 08:1682, 08:1684,
08:1686, 08:1688, 08:1702, 08:1703
\c_hook_GENERIC_<type>/./<place>_tl
..... 245
\c_hook_GENERIC_class/./after_-
tl
08:915
\c_hook_GENERIC_class/./before_-
tl
08:915
\c_hook_GENERIC_cmd/./after_tl .
..... 08:915
\c_hook_GENERIC_cmd/./before_tl
..... 08:915
\c_hook_GENERIC_env/./after_tl .
..... 08:915
\c_hook_GENERIC_env/./before_tl
..... 08:915
\c_hook_GENERIC_env/./begin_tl .
..... 08:915
\c_hook_GENERIC_env/./end_tl 08:915
\c_hook_GENERIC_file/./after_tl
..... 08:915
\c_hook_GENERIC_file/./before_-
tl
08:915
\c_hook_GENERIC_include/./after_-
tl
08:915
\c_hook_GENERIC_include/./before_-
tl
08:915

```

**File Key:** 01=ltchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\c__hook_generic_include./end_-
 tl 08:915
\c__hook_generic_package./after_-
 tl 08:915
\c__hook_generic_package./before_-
 tl 08:915
__hook_generic_parameter:n
 08:1059, 08:1279, 08:1290
__hook_generic_parameter:w
 08:1280, 08:1281
\c__hook_generics_file_prop
 08:891, 08:938
\c__hook_generics_prop
 08:827, 08:859, 08:915, 08:933, 08:935
\c__hook_generics_reversed_i_-_
 prop 08:836, 08:863, 08:938
\c__hook_generics_reversed_iii_-_
 prop 08:840, 08:867, 08:938
__hook_gput_code:nnn
 08:486, 08:491,
 08:497, 08:500, 08:583, 08:584, 08:662
__hook_gput_code_store:nnn
 08:486, 08:513, 08:516
__hook_gput_next_code:nn
 08:2127, 08:2133,
 08:2141, 08:2144, 08:2144, 08:669
__hook_gput_next_do:nn . 08:2150,
 08:2155, 08:2155, 08:2157, 08:2168,
 08:2170, 08:670, 08:691, 08:708, 243
__hook_gput_next_do:Nnn
 08:2172, 08:2175
__hook_gput_undeclared_hook:nnn
 08:651,
 08:651, 08:663, 08:682, 08:702, 243
__hook_gremove_code:nn .. 08:958,
 08:961, 08:962, 08:989, 08:997, 08:1025
__hook_gset_rule:nnnn
 08:1303, 08:1305, 08:1308, 08:1310,
 08:1315, 08:1339, 08:1341, 08:1346
__hook_guess_arg_count:NN 09:467,
 09:467, 09:469, 09:492, 09:494, 09:500
__hook_guess_arg_count:nw
 09:467, 09:480, 09:485, 09:488
__hook_guess_arg_count:wN
 09:467, 09:471, 09:475
\c__hook_hash_tl ... 09:11, 09:254,
 09:255, 09:257, 09:344, 08:1224, 257
\c__hook_hashes_tl . 09:11, 09:179,
 09:180, 09:182, 09:345, 08:1151, 314
\g__hook_hook_curr_name_tl
 08:32, 08:366, 08:376,
 08:417, 08:429, 08:444, 08:445,
 08:452, 08:462, 08:463, 08:484, 237
__hook_hook_gput_code_do:nnn ...
 08:240, 08:257, 08:486,
 08:520, 08:529, 08:591, 08:602, 08:654
__hook_if_cmd_hook:n
 08:2509, 08:2511, 08:2525
__hook_if_cmd_hook:nTF
 08:2509, 08:2527, 08:2068
__hook_if_cmd_hook:w 08:2512, 08:2513
__hook_if_cmd_hook:wTF
 08:2509
__hook_if_cmd_hook_p:n
 08:2509
__hook_if_cmd_hook_p:w
 08:2509
__hook_if_declared:n
 08:2474
__hook_if_declared:nTF . 08:2474,
 09:67, 08:75, 08:93, 08:178, 08:207,
 08:210, 08:297, 08:315, 08:636,
 08:861, 08:1037, 08:1055, 08:2067, 227
__hook_if_declared_p:n
 08:2474
__hook_if_DEPRECATED_GENERIC:n .
 08:2497
__hook_if_DEPRECATED_GENERIC:nTF
 08:2489,
 08:753, 08:807, 08:985, 08:1021,
 08:1312, 08:1343, 08:1836, 08:1954
__hook_if_DEPRECATED_GENERIC:w .
 08:2498, 08:2499
__hook_if_DEPRECATED_GENERIC_-
 p:n 08:2489
__hook_if_DISABLED:n
 08:273
__hook_if_DISABLED:nTF
 08:2146, 08:264, 08:294,
 08:312, 08:524, 08:595, 08:1848,
 08:1860, 08:1943, 08:1966, 08:2046, 233
__hook_if_DISABLED_p:n
 08:264
__hook_if_EXECUTE_IMMEDIATELY:n
 08:2413
__hook_if_EXECUTE_IMMEDIATELY:nTF
 08:2339, 08:2345, 08:2354,
 08:2413, 08:503, 08:586, 08:1319
__hook_if_EXECUTE_IMMEDIATELY_-
 p:n 08:2413
__hook_if_file_hook:w
 08:876, 08:879, 08:881
__hook_if_file_hook:wTF
 08:2318, 08:712, 08:876, 286
__hook_if_file_hook_p:w
 08:876
__hook_if_GENERIC:n
 08:2489
__hook_if_GENERIC:nTF
 08:2489, 08:736,
 08:794, 08:1058, 08:1846, 08:1962
__hook_if_GENERIC:w 08:2490, 08:2491
__hook_if_GENERIC_p:n
 08:2489
__hook_if_GENERIC_reversed:n 08:2529
__hook_if_GENERIC_reversed:nTF .
 08:2529, 08:302, 08:320, 08:748, 08:802

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__hook_if_generic_reversed:w ...
 08:2530, 08:2531
__hook_if_generic_reversed_p:n ...
 08:2529
__hook_if_has_hash:n 09:311
__hook_if_has_hash:nTF
 09:169, 09:244, 09:311, 318
__hook_if_has_hash:w
 09:311, 09:312, 09:313, 09:319, 09:321
__hook_if_has_hash_check:w ...
 09:311, 09:318, 09:323
__hook_if_has_hash_p:n 09:311
__hook_if_label_case:nnnn ...
 08:2083,
 08:1412, 08:1412, 08:1571, 08:1653
__hook_if_public_command:N .. 09:108
__hook_if_public_command:NTF ...
 09:77, 09:87, 311
__hook_if_public_command:w ...
 09:77, 09:111, 09:117
__hook_if_replacing_args: . 08:2543
__hook_if_replacing_args:TF ...
 08:2539, 08:2545, 08:2550,
 08:2551, 08:2556, 08:2557, 08:2562,
 08:505, 08:547, 08:558, 08:634, 08:1071
__hook_if_reversed:n 08:2480
__hook_if_reversed:nTF . 08:2480,
 08:1493, 08:1526, 08:1883, 08:1923,
 08:1925, 08:1984, 08:2023, 08:2026
__hook_if_reversed_p:n 08:2480
__hook_if_structure_exist:n 08:2468
__hook_if_structure_exist:nTF ..
 08:2149, 08:2447, 08:2468, 08:2704,
 08:150, 08:162, 08:964, 08:999, 227
__hook_if_structure_exist_p:n ..
 08:2468
__hook_if_usable:n 08:2462
__hook_if_usable:nTF
 08:2371, 08:2462,
 08:2810, 08:106, 08:518, 08:532,
 08:560, 08:589, 08:605, 08:738,
 08:796, 08:829, 08:981, 08:1017,
 08:1480, 08:1515, 08:1858, 08:1882,
 08:1941, 08:1964, 08:1982, 08:2044, 227
__hook_if_usable_p:n
 08:2462, 08:1916, 08:2016
__hook_if_usable_use:n
 08:2305, 08:2320, 08:2323, 08:2326, 286
__hook_include_legacy_code_-_
 chunk:n
 ... 08:126, 08:142, 08:229, 08:229,
 08:231, 08:249, 08:251, 08:1479, 08:1514
__hook_init_structure:n 08:2159,
 08:119, 08:139, 08:146, 08:146,
 08:148, 08:158, 08:160, 08:539,
 08:613, 08:653, 08:1325, 08:1350, 226
__hook_initialize_all:
 08:2881, 08:1420, 08:1420,
 08:1422, 08:1446, 08:1448, 08:1470
__hook_initialize_hook_code:n ..
 08:2214, 08:2235,
 08:2264, 08:1424, 08:1451, 08:1473,
 08:1473, 08:1475, 08:1507, 08:1509, 258
__hook_initialize_single:NNn ...
 08:1499, 08:1537, 08:1553, 08:1553,
 08:1555, 08:1633, 08:1635, 08:1637, 263
__hook_label_0_t1 08:1548
__hook_label_if_exist_apply:nnnTF
 08:1720, 08:1722, 08:1724, 08:1727, 272
__hook_label_ordered:nn 08:1404, 263
__hook_label_ordered:nnTF
 08:1367, 08:1374, 08:1381, 08:1404, 262
__hook_label_ordered_p:nn . 08:1404
__hook_label_pair:nn
 08:1366, 08:1373,
 08:1380, 08:1385, 08:1390, 08:1395,
 08:1396, 08:1396, 08:1802, 08:1803, 263
__hook_labels_int 08:1548,
 08:1558, 08:1562, 08:1594, 08:1616,
 08:1640, 08:1644, 08:1680, 08:1705, 269
__hook_labels_seq
 08:1548, 08:1557, 08:1563, 08:1581,
 08:1639, 08:1645, 08:1664, 08:1808
__hook_list_if_rule_exists:nnnTF
 .. 08:2074, 08:2094, 08:2095, 08:2097
__hook_list_one_rule:nnn
 .. 08:2074, 08:2085, 08:2086, 08:2092
__hook_list_rules:nn .. 08:2074,
 08:2074, 08:2111, 08:1903, 08:2003, 280
__hook_log:nn 08:1818,
 08:1821, 08:1826, 08:1832, 08:1834,
 08:1839, 08:1950, 08:1952, 08:1957
__hook_log_cmd:n 08:1820, 08:1825,
 08:1829, 08:1831, 08:1843, 08:1961
__hook_log_line:n 08:1818, 08:1828,
 08:1859, 08:1861, 08:1865, 08:1879,
 08:1891, 08:1901, 08:1919, 08:1938,
 08:1965, 08:1967, 08:1971, 08:1979,
 08:1993, 08:2001, 08:2019, 08:2040
__hook_log_line_indent:n
 08:1818, 08:1830,
 08:1869, 08:1875, 08:1885, 08:1892,
 08:1906, 08:1914, 08:1973, 08:1976,
 08:1986, 08:1994, 08:2006, 08:2014
__hook_log_next_code:n . 08:1998,
 08:2053, 08:2053, 08:2058, 08:2060
__hook_log_next_code:w
 08:1897, 08:2056

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__hook_make_name:n 08:357, 08:363, 08:372,
..... 08:378, 08:378, 08:432, 08:475, 236
__hook_make_name:w 08:378, 08:380, 08:384
__hook_make_prefixes:w 09:154, 09:207, 09:282, 09:303, 09:308
__hook_make_usable:n 08:97, 08:133, 08:318, 08:800, 08:833
__hook_make_usable:nn 09:211, 09:503, 08:81,
..... 08:102, 08:102, 08:104, 08:130,
..... 08:132, 08:300, 08:743, 08:746, 229
__hook_misused_if_replacing_-
args:nn .. 08:2539, 08:2540, 08:2546
__hook_msg_pair_found:nnn
..... 08:1745, 08:1752, 08:1759, 08:1767,
..... 08:1775, 08:1786, 08:1799, 08:1799
\g__hook_name_stack_seq
..... 08:32, 08:418, 08:419, 08:423,
..... 08:430, 08:444, 08:451, 08:459, 08:469
__hook_new:n ... 08:89, 08:91, 08:193
__hook_new:nn 08:67, 08:70,
..... 08:72, 08:73, 08:90, 08:181, 08:213
__hook_new_pair:nnn 08:202, 08:204, 08:205
__hook_new_reversed:n 08:189, 08:191
__hook_new_reversed:nn 08:170, 08:173,
..... 08:175, 08:176, 08:190, 08:196, 08:214
__hook_next\langle hook\rangle 226
__hook_next_gset:nn 08:2164, 08:2189, 08:2398, 08:154,
..... 08:970, 08:1089, 08:1095, 08:1109
\c__hook_nine_parameters_t1 08:35,
..... 08:114, 08:1060, 08:1175, 08:1874
__hook_normalise_code_pool:n ...
..... 08:120, 08:1136,
..... 08:1136, 08:1138, 08:1188, 08:1190, 229
__hook_normalise_cs_args:nn ...
..... 08:116, 08:117, 08:1112,
..... 08:1112, 08:1114, 08:1132, 08:1134, 229
__hook_normalise_fn:nn 08:564, 08:1142, 08:1161, 08:1186, 255
__hook_normalize_hook_args:Nn ..
..... 08:2127, 08:2133, 08:2140,
..... 08:2185, 08:2340, 08:2346, 08:2355,
..... 08:2840, 08:70, 08:72, 08:89, 08:173,
..... 08:175, 08:189, 08:267, 08:291,
..... 08:385, 08:394, 08:1821, 08:1826
__hook_normalize_hook_args:Nnn ..
..... 08:202, 08:204, 08:385,
..... 08:399, 08:491, 08:497, 08:582, 08:961
__hook_normalize_hook_args_-aux:Nn .. 08:385, 08:385, 08:396, 08:401, 08:409
__hook_normalize_hook_rule_-
args:Nnnnn .. 08:385, 08:407, 08:1305
\l__hook_param_text_t1
..... 09:8, 09:175,
..... 09:196, 09:250, 09:271, 09:300, 317
__hook_parameter:n 08:1100, 08:1158, 08:1270, 08:1270,
..... 08:1272, 08:1287, 08:1289, 08:1853
\c__hook_parameter_cmd 08:950, 08:955
\c__hook_parameter_cmd./after_-
t1 08:950
\c__hook_parameter_cmd./before_-
t1 08:950
__hook_parse_dot_label:nN
..... 08:334, 08:336, 08:336
__hook_parse_dot_label:w
..... 08:336, 08:348, 08:351
__hook_parse_dot_label_aux:w ...
..... 08:336, 08:354, 08:362
__hook_parse_dot_label_cleanup:w ...
..... 08:336, 08:358, 08:361
__hook_parse_label_default:nN ..
..... 08:330, 08:330, 08:397,
..... 08:403, 08:404, 08:411, 08:412, 08:414
__hook_patch_check:Nnn
..... 09:77, 09:81, 09:84, 09:87, 09:97
__hook_patch_cmd_or_delay:Nnn ..
..... 09:29, 09:42, 09:46, 09:46, 09:56, 310
__hook_patch_command:Nnn
..... 09:56, 09:74, 09:77, 09:77, 09:77, 310
__hook_patch_debug:n
..... 09:18, 09:18, 09:79, 09:80,
..... 09:83, 09:86, 09:89, 09:158, 09:233,
..... 09:372, 09:407, 09:410, 09:411,
..... 09:416, 09:426, 09:429, 09:430, 09:435
__hook_patch_DeclareRobustCommand:Nnn
..... 09:126, 09:130, 09:130, 312
__hook_patch_DeclareRobustCommand_-
aux:Nnn 09:132, 09:135
__hook_patch_expand_redefine:NNnn
..... 09:141,
..... 09:146, 09:151, 09:154, 09:154,
..... 09:156, 09:229, 09:231, 09:370, 312
__hook_patch_newcommand:Nnn ...
..... 09:127, 09:140, 09:144, 09:144, 312
\l__hook_patch_num_args_int
..... 09:7, 09:159,
..... 09:164, 09:167, 09:181, 09:211,
..... 09:220, 09:234, 09:239, 09:242,
..... 09:256, 09:500, 09:503, 09:523, 09:530

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\l_hook_patch_prefixes_tl
 ... 09:8, 09:206, 09:281, 09:298, 317
\l_hook_patch_required_catcodes:
 09:441,
 09:441, 09:462, 09:538, 09:566, 325
\l_hook_patch_retokenize:Nnnn ...
 09:412, 09:431, 09:496,
 09:496, 09:498, 09:543, 09:545, 321
\l_hook_post_initialization_-
 defs: . 08:2291, 08:2291, 08:2293,
 08:2298, 08:2301, 08:2303, 08:1443
\l_hook_preamble_hook:n
 08:2196, 08:2200, 08:2211,
 08:2224, 08:2234, 08:2244, 08:2263,
 08:2272, 08:2297, 08:2330, 08:2365,
 08:2384, 08:1466, 08:1842, 08:1960, 283
\l_hook_print_args:n
\l_hook_print_args:nn 08:1850, 08:2063
\l_hook_prop_gput_labeled_-
 cleanup:nnn . 08:486, 08:545, 08:555
\l_hook_prop_gput_labeled_do:Nnn
 08:569, 08:572
\l_hook_prop_gput_labeled_-
 do:Nnnn 08:486
\l_hook_rear_tl
 08:1548, 08:1579, 08:1585,
 08:1586, 08:1609, 08:1610, 08:1662,
 08:1670, 08:1671, 08:1698, 08:1699
\l_hook redefine_with_hooks:Nnnn
 .. 09:154, 09:214, 09:286, 09:296, 316
\l_hook_replace_text_tl
 09:8, 09:176, 09:183,
 09:184, 09:185, 09:190, 09:197,
 09:224, 09:251, 09:258, 09:259,
 09:260, 09:265, 09:272, 09:292, 314
\l_hook_replacement_spec:N
 08:1196, 08:1202
\l_hook_replacing_args_false: ...
 08:2126, 08:2367, 08:2539, 08:2553,
 08:237, 08:490, 08:641, 08:1429, 239
\l_hook_replacing_args_reset: ...
 08:2128, 08:2134, 08:2370, 08:2539,
 08:2559, 08:243, 08:492, 08:498, 08:1432
\l_hook_replacing_args_true: ...
 08:2132,
 08:2539, 08:2547, 08:496, 08:1430
\g_hook_replacing_stack_seq 08:2539
\l_hook_retokenize_patch:Nnn ...
 09:92, 09:367, 09:367
\l_hook_return_tl
 .. 08:2561, 08:2562, 08:25, 08:451,
 08:452, 08:459, 08:463, 08:557,
 08:565, 08:570, 08:574, 08:575,
 08:620, 08:623, 08:977, 08:1012,
 08:1595, 08:1596, 08:1682, 08:1683
\l_hook_rollback_tidying: ... 08:2921
\l_hook_rule_<_gset:nnn
\l_hook_rule_>_gset:nnn
\l_hook_rule_after_gset:nnn ...
 08:1363, 08:1370, 08:1376
\l_hook_rule_before_gset:nnn ...
 08:1363, 08:1363, 08:1369, 268
\l_hook_rule_gclear:nnn
 08:1326, 08:1351, 08:1393, 08:1394, 262
\l_hook_rule_incompatible_error_-
 gset:nnn 08:1383
\l_hook_rule_incompatible_warning_-
 gset:nnn 08:1383
\l_hook_rule_unrelated_gset:nnn .
 08:1393, 08:1393, 262
\l_hook_rule_voids_gset:nnn ...
 08:1377, 08:1377
\l_hook_seq_cservername: ...
 08:1546,
 08:1547, 08:1565, 08:1599, 08:1647,
 08:1688, 08:1748, 08:1755, 08:1813
\l_hook_set_default_hook_label:n
 08:2773, 08:467, 08:467
\l_hook_set_default_label:n ...
 08:467, 08:475, 08:477
\l_hook_set_normalise_fn:nn ...
 08:562, 08:1136, 08:1140, 08:1145, 255
\l_hook_str_compare:nn ...
 08:23, 08:23, 08:1398, 08:1406, 08:1415
\l_hook_strip_double_slash:n ...
 08:899, 08:905, 08:906
\l_hook_strip_double_slash:w ...
 08:899, 08:907, 08:908, 08:912
\l_hook_tlname: ...
 08:1546, 08:1546, 08:1552, 08:1564,
 08:1580, 08:1583, 08:1585, 08:1589,
 08:1601, 08:1603, 08:1606, 08:1609,
 08:1614, 08:1646, 08:1663, 08:1667,
 08:1669, 08:1674, 08:1690, 08:1692,
 08:1695, 08:1697, 08:1703, 08:1746,
 08:1747, 08:1753, 08:1754, 08:1812
\l_hook_tlgclear:N
 08:2407,
 08:2408, 08:2409, 08:64, 08:64,
 08:66, 08:244, 08:259, 08:1004,
 08:1005, 08:1009, 08:1590, 08:1675
\l_hook_tlgput:Nn 08:1494, 08:1496,
 08:1527, 08:1531, 08:1596, 08:1623,
 08:1683, 08:1712, 08:1718, 08:1718, 270
\l_hook_tlgput_left:Nn ...
 08:58, 08:58, 08:1494, 08:1528
\l_hook_tlgput_right:Nn 08:2181,
 08:55, 08:55, 08:57, 08:614, 08:1496,
 08:1532, 08:1625, 08:1714, 08:1716

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__hook_tl_gset:Nn 08:2180, 08:2394, 08:2932,
08:49, 08:49, 08:51, 08:53, 08:54,
08:56, 08:60, 08:1365, 08:1372,
08:1379, 08:1384, 08:1389, 08:1519
__hook_tl_gset_eq:NN 08:63, 08:63, 08:65
__hook_tl_set:Nn 08:47,
08:47, 08:1151, 08:1564, 08:1646, 224
__hook_tmp:w 08:2078, 08:2100, 08:2109,
08:2120, 08:2930, 08:2947, 08:2948,
09:173, 09:179, 09:180, 09:182,
09:248, 09:254, 09:255, 09:257,
09:453, 09:456, 09:460, 09:512,
09:515, 09:536, 09:548, 09:551,
09:564, 08:34, 08:34, 08:420, 08:424,
08:426, 08:1147, 08:1158, 08:1174,
08:1180, 08:1183, 08:1873, 08:1876, 324
\l__hook_tmfa_bool 08:24,
08:1902, 08:1905, 08:1913, 08:1922,
08:2002, 08:2005, 08:2013, 08:2022, 276
\l__hook_tmfa_tl .. 09:166, 09:168,
09:170, 09:241, 09:243, 09:245,
09:373, 09:388, 09:391, 09:459,
09:462, 09:504, 09:508, 09:524,
09:531, 09:535, 09:538, 09:563,
09:566, 08:25, 08:430, 08:1159, 08:1181
\l__hook_tmfp_tl .. 09:378, 09:389,
09:392, 08:25, 08:1148, 08:1155, 08:1183
__hook_toplevel\hook 251
__hook_toplevel_gset:nn 08:2399, 08:153, 08:969,
08:974, 08:1089, 08:1093, 08:1108
__hook_try_declar ing_generic_-
hook:nnn 08:526,
08:597, 08:656, 08:657, 08:659,
08:675, 08:677, 08:696, 08:699, 243
__hook_try_declar ing_generic_-
hook:nNnn 08:701, 08:707, 08:710, 08:710, 243
__hook_try_declar ing_generic_-
hook:wn .. 08:730, 08:733, 08:788,
08:791, 08:818, 08:821, 08:850, 08:853
__hook_try_declar ing_generic_-
hook:wNTF 08:661, 08:668, 08:679,
08:688, 08:723, 08:729, 08:782, 246
__hook_try_declar ing_generic_-
hook_split:nNnn 08:710, 08:715, 08:718, 08:721
__hook_try_declar ing_generic_-
next_hook:nn 08:2151,
08:656, 08:666, 08:686, 08:705, 243
__hook_try_file_hook:n 08:2305, 08:2313, 08:2316, 286
__hook_try_patch_with_catcodes:Nnnw
... 09:385, 09:402, 09:402, 09:404,
09:417, 09:421, 09:423, 09:436, 320
__hook_try_put_cmd_hook:n
..... 09:21, 09:21, 09:23,
09:32, 09:34, 08:742, 08:799, 08:832
__hook_try_put_cmd_hook:w
..... 09:21, 09:24, 09:25, 09:35, 09:36
__hook_unpatchable_cases:n
..... 09:591, 09:593
__hook_update_hook_code:n
..... 08:2162, 08:2178, 09:226,
09:540, 08:82, 08:303, 08:321,
08:521, 08:592, 08:982, 08:1018,
08:1330, 08:1355, 08:1419, 08:1419,
08:1424, 08:1431, 08:1450, 08:1454, 264
__hook_use:wn .. 08:2247, 08:2261,
08:2305, 08:2305, 08:2308, 08:2310, 286
__hook_use_end:
..... 08:2255, 08:2258, 08:2265
__hook_use_i_delimit_by_s_-
mark:nw 08:45, 08:46, 08:1222
__hook_use_initialized:n 08:2196,
08:2201, 08:2203, 08:2228, 08:2249,
08:2295, 08:2372, 08:2386, 08:1465, 283
__hook_use_initialized:nnw
.. 08:2268, 08:2273, 08:2275, 08:2296
__hook_use_none_delimit_by_s_-
mark:w 08:2415, 08:2482,
09:489, 08:45, 08:45, 08:1317, 08:1323
__hook_use_once:n .. 08:2355, 08:2382
__hook_use_once:nn
..... 08:2340, 08:2346, 08:2361,
08:2361, 08:2363, 08:2380, 08:2389
__hook_use_once_clear:n
..... 08:2369, 08:2387, 08:2391,
08:2391, 08:2395, 08:2403, 08:2405, 288
__hook_use_once_set:n
08:2366, 08:2385, 08:2391, 08:2393, 288
__hook_use_undefined:w
..... 08:2253, 08:2257
\g__hook_used_prop
..... 08:31, 08:1425, 08:1437,
08:1452, 08:1458, 08:1502, 08:1540
\l__hook_work_prop 08:30,
08:565, 08:1141, 08:1143, 08:1177,
08:1498, 08:1535, 08:1560, 08:1567,
08:1569, 08:1578, 08:1595, 08:1620,
08:1642, 08:1649, 08:1651, 08:1661,
08:1681, 08:1709, 08:1780, 08:1791, 268
hook\?? internal commands:
__hook\?? 260

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspac e.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur e.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lt hyphen.dtx, 57=ltfinal.dtx

**hook\<hook>** internal commands:  
 \\_\\_hook\<hook> ..... 265  
 \hookleftarrow ..... 30:487  
 \hookrightarrow ..... 30:485  
**hook ??** internal commands:  
 \\_\\_hook~?? ..... 08:1292  
 \phantom ..... 38:75  
 \hrule ..... 18:397, 18:405, 18:431,  
     18:439, 21:289, 21:294, 30:347,  
     30:625, 40:202, 40:207, 40:280,  
     40:290, 41:359, 41:376, 42:589,  
     42:599, 45:395, 02:502, 02:546, 1268  
 \hrulefill ..... 06:884, 06:905, 02:546  
 \hsize ..... 1237  
 \hskip ..... 1224  
 \hskip... ..... 1233  
 \hspace ..... 18:512, 18:516, 18:522, 439  
 \hss ..... 1079  
 \Hwithstroke ..... 21:494, 21:1213, 1248  
 \hwithstroke ..... 21:510, 21:1214, 1248  
 \hyphenation ..... 21:205, 1220  
 \hyphenchar ..... 37:564, 06:842, 06:845,  
     06:854, 06:857, 06:860, 06:865, 1246  
 \hyphenpenalty ..... 24:687, 24:719, 02:321

**I**

\I ..... 50:1335, 50:1467,  
     50:1557, 50:1577, 57:255, 57:550, 02:437  
 \i ..... 21:247, 21:402, 21:452, 21:453,  
     21:454, 21:455, 21:456, 21:457,  
     21:547, 21:587, 21:588, 21:680,  
     21:682, 21:684, 21:686, 21:778,  
     21:1140, 21:1295, 21:1297, 21:1299,  
     21:1301, 21:1352, 21:1355, 21:1358,  
     21:1361, 21:1431, 57:259, 57:554, 1239  
 \ialign ..... 30:344,  
     30:466, 30:537, 30:540, 30:544,  
     30:547, 38:168, 38:170, 38:189,  
     41:191, 42:141, 02:534, 02:536, 1198  
 \IeC ..... 57:357, 57:361, 57:468  
 \if ..... 1232  
 if commands:  
 \if:w .....  
     08:2416, 08:2430, 08:2483, 08:2515,  
     09:343, 08:1036, 08:1196, 08:1221, 288  
 \if\_case:w .....  
     .. 07:1458, 08:1247, 08:1398, 08:1415  
 \if\_catcode:w ..... 09:352, 319  
 \if\_charcode:w ..... 08:2533, 07:946  
 \if\_cs\_exist:w ..... 08:2099,  
     08:2205, 08:2213, 08:2230, 08:2251,  
     08:2278, 08:2417, 08:767, 08:1729  
 \if\_false: .....  
     .... 07:876, 07:879, 07:2399, 07:2403

\if\_int\_compare:w ..... 16:101, 08:1406  
 \if\_meaning:w .....  
     09:340, 09:344, 09:345, 52:241, 07:1303  
 \if\_mode... ..... 413  
 \if\_mode\_horizontal: ..... 16:32, 16:64  
 \if... ..... 1027  
 \IfBlankF ..... 07:2987  
 \IfBlankT ..... 07:2987  
 \IfBlankTF ..... 47:19, 07:2987, 1261  
 \IfBold ..... 650  
 \IfBooleanF ..... 22:130, 22:138, 07:2962  
 \IfBooleanT ..... 07:2962  
 \IfBooleanTF ..... 17:32,  
     35:67, 35:69, 35:156, 07:2962, 191  
 \IfClassAtLeastF ..... 50:283  
 \IfClassAtLeastT ..... 50:283  
 \IfClassAtLeastTF ..... 977  
 \IfClassAtLeastTF .....  
     50:165, 50:293, 50:294, 50:322, 50:323  
 \IfClassLoadedF ..... 50:283  
 \IfClassLoadedT ..... 50:283  
 \IfClassLoadedTF ..... 977  
 \IfClassLoadedTF .....  
     50:263, 50:289, 50:290, 50:318, 50:319  
 \IfClassLoadedWithOptionsF ..... 50:283  
 \IfClassLoadedWithOptionsT ..... 50:301, 50:330  
 \IfClassLoadedWithOptionsTF ..... 977  
 \IfClassLoadedWithOptionsTF .....  
     .... 50:263, 50:283  
 \ifcoremisses ..... 33:1222, 33:1413  
 \ifcsname ..... 24:340,  
     25:440, 25:443, 25:653, 25:656,  
     26:128, 28:153, 29:42, 29:53, 29:75,  
     29:86, 33:1248, 50:1220, 50:1355,  
     53:433, 06:740, 06:757, 54:1907, 569  
 \ifdefined ..... 02:570, 02:634,  
     29:623, 05:22, 05:71, 05:72, 05:73,  
     05:74, 05:117, 05:118, 05:119, 57:319  
 \IfDocumentMetadataTF ..... 17:5, 17:17  
 \IfExplAtLeastTF ..... 78  
 \IfExplAtLeastTF ..... 05:198, 05:205, 05:207  
 \iff ..... 30:507  
 \IfFileAtLeastF ..... 50:283  
 \IfFileAtLeastT ..... 50:283  
 \IfFileAtLeastTF ..... 977  
 \IfFileAtLeastTF .....  
     50:165,  
     50:295, 50:296, 50:324, 50:325, 1264  
 \IfExists ..... 442, 978  
 \IfExists ..... 20:469,  
     20:481, 20:617, 20:673, 20:698,  
     01:162, 05:37, 05:68, 05:114, 50:925,  
     52:149, 52:185, 52:195, 57:651, 1205  
 \IfFileLoadedF ..... 50:303  
 \IfFileLoadedT ..... 50:303

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltypen.dtx, 57=ltfinal.dtx

\IfFileLoadedTF . . . . . 50:303, 1268  
 \iffontchar . . . . . 21:866, 21:1060, 21:1165, 21:1167,  
   21:1169, 21:1216, 33:1218, 33:1219  
 \IfFontSeriesContextTF . . . . .  
   . . . . . 29:496, 29:532, 29:534, 650  
 \IfFormatAtLeastF . . . . . 50:283  
 \IfFormatAtLeastT . . . . . 50:283  
 \IfFormatAtLeastTF . . . . . 977  
 \IfFormatAtLeastTF . . . . . 50:165,  
   50:297, 50:298, 50:326, 50:327, 1047  
 \IfHookEmptyF . . . . . 08:2807, 205  
 \IfHookEmptyT . . . . . 08:2807, 1270  
 \IfHookEmptyTF . . . . .  
   . . . . . 08:2807, 08:2912, 37:349, 37:366, 200  
 \IfHookExistsTF . . . . . 08:2810, 08:2911, 299  
 \ifincsname . . . . . 18:475, 1033  
 \ifinner . . . . . 18:239, 18:290, 38:277, 38:285,  
   38:305, 38:332, 45:57, 45:126, 45:315  
 \IfInstanceExistsF . . . . . 11:1186, 348  
 \IfInstanceExistsT . . . . . 11:1186, 348  
 \IfInstanceExistsTF . . . . . 11:1186, 348  
 \IfLabelExistsF . . . . . 36:181, 1270  
 \IfLabelExistsT . . . . . 36:181, 750  
 \IfLabelExistsTF . . . . . 36:181, 36:275, 1268  
 \IfLabelExistTF . . . . . 1268  
 \IfMarksEqualF . . . . . 48:372, 945  
 \IfMarksEqualT . . . . . 48:372, 945  
 \IfMarksEqualTF . . . . . 48:372, 48:487, 945  
 \ifmmode . . . . . 1193  
 \IfNoValueF . . . . . 07:2981  
 \IfNoValueT . . . . . 07:2981  
 \IfNoValueTF . . . . .  
   36:120, 57:629, 57:636, 57:643, 07:2981  
 \ifnum . . . . . 1064  
 \ifodd . . . . . 28:1156,  
   42:320, 42:344, 42:378, 42:400,  
   45:68, 45:137, 54:21, 54:140, 54:614,  
   54:672, 54:986, 54:989, 54:1022,  
   54:1025, 54:1136, 54:1139, 54:1298,  
   54:1301, 54:1578, 54:1581, 54:1699,  
   54:1702, 54:1822, 54:2077, 54:2085  
 \IfPackageAtLeastF . . . . . 50:283  
 \IfPackageAtLeastT . . . . . 50:283  
 \IfPackageAtLeastTF . . . . . 977  
 \IfPackageAtLeastTF . . . . .  
   50:165, 50:291, 50:292, 50:320, 50:321  
 \IfPackageLoadedF . . . . . 50:283  
 \IfPackageLoadedT . . . . . 50:283  
 \IfPackageLoadedTF . . . . . 977  
 \IfPackageLoadedTF . . . . .  
   50:263, 50:287, 50:288, 50:316, 50:317  
 \IfPackageLoadedWithOptionsF . . . . . 50:283  
 \IfPackageLoadedWithOptionsT . . . . . 50:283  
 \IfPackageLoadedWithOptionsTF . . . . . 977  
 \IfPackageLoadedWithOptionsTF . . . . .  
   50:263, 50:299, 50:300, 50:328, 50:329  
 \IfPDFManagementActiveTF . . . . . 57:710, 1192  
 \IfPropertyExistsF . . . . . 36:166, 750  
 \IfPropertyExistsT . . . . . 36:166, 750  
 \IfPropertyExistsTF . . . . . 36:166, 36:274, 750  
 \IfPropertyExistTF . . . . . 1268  
 \IfPropertyRecordedF . . . . . 36:196, 1270  
 \IfPropertyRecordedT . . . . . 36:196, 750  
 \IfPropertyRecordedTF . . . . . 36:196, 36:276, 749  
 \ifsafesubencodingfound . . . . .  
   . . . . . 33:1221, 33:1235, 33:1240, 33:1409  
 \IfTargetDateBefore . . . . . 50:1793  
 \ifthenelse . . . . . 741  
 \IfValueF . . . . . 07:2984  
 \IfValueT . . . . . 07:2984  
 \IfValueTF . . . . . 07:2984  
 \Ifvbox . . . . . 54:321, 54:378, 54:425, 54:506, 54:522  
 \ifvoid . . . . . 1234  
 \ifx . . . . . 1251  
 \ignorespaces . . . . . 18:50, 18:144, 18:163,  
   18:175, 18:186, 18:202, 18:215,  
   18:559, 20:71, 20:139, 20:194, 21:72,  
   24:292, 24:302, 24:312, 24:358,  
   28:301, 28:340, 37:339, 37:356,  
   37:372, 37:381, 37:430, 37:435,  
   37:441, 38:313, 38:340, 38:445,  
   38:446, 39:55, 39:241, 40:193,  
   40:490, 40:507, 40:524, 41:57,  
   41:62, 41:68, 41:83, 41:92, 41:105,  
   41:109, 41:116, 41:123, 41:125,  
   41:134, 41:154, 41:237, 41:301,  
   41:303, 41:305, 41:332, 42:36, 42:46,  
   42:66, 42:75, 42:109, 42:120, 42:131,  
   42:143, 42:148, 42:154, 43:43, 43:45,  
   44:110, 45:17, 45:24, 45:477, 45:495,  
   45:513, 47:7, 47:9, 53:330, 1230  
 \ignorespacesafterend . . . . . 37:7  
 \IJ . . . . . 21:250,  
   21:436, 21:550, 21:1141, 57:650, 1241  
 \ij . . . . . 21:249,  
   21:434, 21:549, 21:1142, 57:650, 1259  
 \Im . . . . . 30:321  
 \imath . . . . . 30:316  
 \immediate . . . . . 1196  
 \in . . . . . 30:436, 30:468  
 in commands:  
   in\_callback . . . . . 04:959  
 \in\_callback . . . . . 46  
 \include . . . . . 442  
 \include . . . . . 20:218, 20:266, 20:268,  
   20:284, 20:286, 20:345, 20:397, 1039  
 include/.../after . . . . . 1039

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

include/.../before ..... 1039  
 include/.../end ..... 1039  
 include/after ..... 1039  
 include/before ..... 1039  
 include/end ..... 1039  
 \IncludeInRelease .. 08:2122, 08:2137,  
     08:2155, 08:2168, 08:2186, 08:2191,  
     08:2196, 08:2218, 08:2238, 02:581,  
     08:2268, 08:2286, 08:2291, 08:2301,  
     08:2305, 08:2308, 08:2335, 08:2350,  
     08:2361, 08:2380, 08:2391, 08:2403,  
     08:2426, 08:2443, 08:2509, 08:2525,  
     02:615, 08:2582, 08:2611, 02:622,  
     08:2719, 08:2728, 08:2734, 08:2741,  
     08:2746, 08:2753, 08:2757, 08:2764,  
     08:2784, 08:2791, 02:648, 08:2885,  
     09:21, 09:32, 09:154, 02:688, 02:693,  
     09:229, 02:703, 09:402, 09:421,  
     09:467, 09:492, 09:496, 09:543,  
     09:570, 09:573, 09:603, 10:179,  
     03:71, 11:1196, 03:180, 03:184,  
     03:188, 03:192, 14:138, 14:158,  
     14:192, 14:205, 16:12, 16:45, 16:167,  
     17:20, 17:44, 17:58, 18:5, 18:22,  
     04:3, 18:55, 18:66, 18:82, 18:87,  
     18:99, 18:105, 18:133, 18:153,  
     18:167, 18:179, 18:192, 18:207,  
     18:236, 18:251, 18:267, 18:286,  
     18:320, 18:354, 18:380, 18:413,  
     18:471, 18:482, 18:489, 18:495,  
     18:501, 18:507, 18:515, 18:521,  
     18:535, 18:541, 20:10, 20:84, 20:142,  
     20:220, 20:257, 20:278, 20:293,  
     20:354, 20:403, 20:464, 20:474,  
     20:500, 20:532, 20:554, 20:572,  
     20:582, 20:598, 20:623, 20:628,  
     20:636, 20:650, 20:661, 20:677,  
     20:710, 21:77, 21:104, 21:148,  
     21:169, 21:324, 21:332, 21:353,  
     21:369, 04:235, 04:258, 04:281,  
     22:24, 22:30, 22:38, 22:59, 22:78,  
     22:88, 22:94, 22:117, 22:125, 22:147,  
     22:174, 22:209, 22:225, 22:233,  
     22:251, 23:5, 23:11, 23:19, 23:26,  
     24:24, 24:53, 24:210, 24:232, 24:284,  
     24:295, 24:305, 24:354, 24:367,  
     24:445, 24:456, 24:464, 24:496,  
     24:504, 24:518, 24:539, 24:575,  
     24:662, 24:725, 25:2, 25:385, 25:392,  
     25:404, 25:416, 25:424, 25:506,  
     25:522, 25:600, 25:613, 25:623,  
     25:630, 25:638, 25:710, 25:729,  
     25:735, 25:739, 26:113, 26:158,  
     26:161, 26:541, 26:550, 27:2, 27:22,  
     28:49, 28:78, 28:138, 28:180, 28:210,  
     28:241, 28:275, 28:307, 28:398,  
     28:411, 28:417, 28:457, 28:505,  
     28:532, 28:557, 28:790, 28:838,  
     28:884, 28:893, 28:1077, 28:1088,  
     29:33, 29:68, 29:101, 29:126, 29:228,  
     29:252, 29:301, 29:348, 29:377,  
     29:395, 29:435, 29:469, 29:498,  
     29:531, 29:543, 29:577, 29:590,  
     29:621, 29:630, 29:655, 29:672,  
     29:688, 29:705, 30:71, 30:89, 30:108,  
     30:117, 30:629, 30:641, 32:27, 32:34,  
     33:607, 05:2, 05:11, 05:20, 35:12,  
     35:29, 35:46, 35:62, 35:76, 35:96,  
     35:111, 35:115, 35:139, 35:148,  
     35:159, 35:169, 35:181, 35:192,  
     05:58, 36:264, 37:10, 37:76, 37:140,  
     37:182, 37:187, 05:112, 37:199,  
     37:226, 37:245, 37:267, 37:295,  
     37:312, 37:334, 37:345, 37:361,  
     37:375, 37:383, 05:133, 37:394,  
     37:402, 37:415, 37:421, 37:433,  
     37:438, 37:448, 37:467, 37:484,  
     37:500, 37:511, 05:148, 37:525,  
     37:547, 37:574, 37:590, 37:603,  
     37:617, 37:629, 37:635, 37:641,  
     37:648, 37:666, 37:679, 37:695,  
     37:718, 37:728, 05:159, 38:79, 38:87,  
     05:169, 38:109, 38:118, 38:137,  
     38:144, 38:152, 38:157, 38:165,  
     38:176, 38:216, 38:233, 38:272,  
     05:178, 38:280, 38:290, 38:317,  
     38:399, 05:192, 38:408, 38:441,  
     38:449, 38:462, 38:474, 38:486,  
     05:199, 38:495, 05:204, 06:5, 06:13,  
     39:125, 39:135, 39:143, 39:155,  
     40:4, 40:14, 40:42, 40:58, 40:71,  
     40:91, 40:112, 40:122, 40:136,  
     40:142, 40:147, 40:156, 40:164,  
     40:220, 40:228, 40:241, 40:261,  
     40:299, 40:308, 40:327, 40:355,  
     40:378, 40:400, 40:477, 40:495,  
     40:511, 40:531, 40:537, 40:557,  
     40:565, 40:595, 40:606, 06:56, 06:62,  
     41:60, 41:65, 41:137, 41:157, 41:222,  
     41:227, 42:10, 42:16, 42:26, 42:39,  
     42:55, 42:69, 42:80, 42:89, 42:100,  
     42:112, 42:146, 42:151, 42:160,  
     42:172, 42:235, 42:250, 42:311,  
     42:369, 42:455, 42:463, 42:472,  
     42:500, 42:530, 42:557, 42:571,  
     42:584, 42:595, 42:607, 42:624,  
     42:644, 42:651, 42:685, 42:745,  
     42:803, 42:819, 01:274, 43:28, 43:34,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

43:50, 43:57, 44:5, 44:20, 44:161,  
 44:168, 44:174, 44:186, 44:193,  
 44:202, 44:225, 45:35, 45:105,  
 06:218, 06:225, 45:206, 45:232,  
 45:280, 45:294, 45:335, 45:352,  
 45:406, 45:412, 45:420, 45:427,  
 45:434, 45:440, 45:446, 45:463,  
 45:482, 45:499, 45:549, 45:559,  
 47:14, 47:25, 47:61, 47:77, 01:21,  
 06:284, 01:291, 06:326, 48:479,  
 49:20, 49:48, 49:69, 50:18, 50:23,  
 50:36, 06:354, 50:51, 50:63, 50:87,  
 50:95, 50:101, 50:126, 50:144,  
 50:167, 50:175, 50:187, 50:200,  
 50:226, 50:245, 50:265, 50:273,  
 50:285, 06:382, 50:313, 50:345,  
 50:361, 50:375, 06:389, 50:393,  
 50:406, 50:422, 50:442, 50:465,  
 50:481, 50:510, 50:524, 06:406,  
 50:556, 50:574, 50:594, 50:612,  
 50:631, 50:641, 50:651, 50:663,  
 50:695, 06:422, 50:706, 50:720,  
 50:730, 50:776, 50:790, 50:805,  
 50:835, 50:862, 50:894, 50:1046,  
 50:1126, 50:1133, 50:1144, 50:1158,  
 50:1208, 06:460, 06:472, 50:1343,  
 50:1474, 06:502, 06:513, 06:538,  
 52:5, 52:15, 52:27, 06:546, 52:89,  
 52:144, 52:181, 52:192, 06:558,  
 52:205, 52:249, 52:257, 06:565,  
 52:268, 52:299, 52:322, 52:343,  
 52:357, 52:365, 52:388, 52:408,  
 52:433, 52:479, 52:495, 52:506,  
 52:533, 53:3, 53:430, 53:470,  
 53:484, 54:24, 54:54, 06:664, 54:153,  
 54:181, 06:681, 01:26, 54:347,  
 54:368, 54:373, 54:421, 54:595,  
 54:655, 54:798, 54:816, 06:730,  
 54:877, 54:898, 06:736, 54:934,  
 54:958, 54:1070, 06:766, 54:1221,  
 54:1390, 54:1472, 54:1566, 54:1688,  
 06:807, 54:1902, 54:1925, 54:1939,  
 54:1967, 06:826, 06:837, 54:2198,  
 54:2216, 54:2235, 54:2281, 06:854,  
 06:865, 06:876, 06:896, 55:244,  
 57:8, 57:16, 57:23, 57:30, 57:37,  
 57:52, 57:71, 57:80, 57:87, 57:106,  
 57:143, 57:166, 57:199, 57:288,  
 57:293, 57:314, 57:339, 57:348,  
 57:441, 02:49, 07:224, 07:238,  
 02:88, 02:103, 02:119, 02:125,  
 02:134, 02:139, 02:148, 07:1091,  
 02:154, 07:1143, 07:1146, 07:1161,  
 07:1174, 07:1189, 02:168, 07:1307,  
 07:1310, 02:182, 07:1339, 07:1342,  
 07:1351, 07:1359, 07:1362, 02:186,  
 07:1523, 02:220, 02:228, 02:233,  
 02:244, 07:1978, 07:1994, 02:289,  
 07:2840, 07:2893, 02:351, 07:2987,  
 07:2993, 07:3009, 02:361, 08:67,  
 08:86, 08:102, 08:130, 08:146,  
 08:158, 08:170, 08:186, 08:199,  
 08:219, 08:229, 08:249, 08:264,  
 08:282, 08:288, 08:308, 08:326,  
 08:486, 08:579, 02:454, 08:630,  
 08:647, 08:656, 08:674, 08:695,  
 08:729, 08:787, 08:817, 08:849,  
 08:876, 08:879, 08:899, 08:902,  
 08:915, 08:933, 08:938, 08:941,  
 08:950, 08:955, 08:958, 08:995,  
 02:472, 08:1032, 08:1082, 08:1089,  
 08:1105, 08:1112, 08:1132, 08:1136,  
 08:1188, 08:1192, 08:1208, 08:1212,  
 08:1240, 08:1243, 08:1266, 08:1270,  
 08:1287, 08:1295, 08:1299, 08:1308,  
 08:1339, 08:1420, 08:1446, 08:1473,  
 08:1507, 08:1553, 08:1635, 08:1832,  
 08:1950, 02:557, 08:2053, 08:2058, 1257  
 \includeonly ..... 442  
 \includeonly ..... 20:218,  
 20:258, 20:260, 20:279, 20:280, 1039  
 \indent ..... 18:538, 39:185, 41:81, 1233  
 \IndentBox ..... 16:91, 16:171, 406  
 \index ..... 934  
 \index 44:182, 46:6, 46:18, 48:252, 49:25,  
 49:36, 49:53, 49:61, 54:624, 54:683, 943  
 \indexentry ..... 46:15  
 \inf ..... 38:25  
 \infty ..... 30:323  
 \initcatcodetable ..... 04:91  
 \input ..... 442, 978  
 \input ... 04:18, 20:633, 01:158, 01:161,  
 26:16, 27:106, 29:751, 29:761,  
 29:771, 30:10, 30:11, 30:12, 30:13,  
 30:14, 30:23, 30:41, 30:42, 30:46,  
 30:47, 30:48, 30:49, 30:54, 30:143,  
 30:144, 30:145, 30:146, 30:661,  
 30:662, 30:663, 01:218, 33:1126,  
 05:108, 05:130, 06:22, 50:680,  
 57:164, 57:178, 57:203, 57:281,  
 57:325, 57:405, 57:656, 01:52, 1228  
 \input@path ..... 6, 1  
 input@path commands:  
     \input@path: ..... 01:223  
 \inputencodingname ..... 57:382, 57:404, 57:486, 1247  
 \InputIfExists ..... 442, 978

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\InputIfFileExists 17:7, 20:616, 20:639,
 20:654, 20:664, 20:674, 20:728,
 21:1546, 24:416, 29:743, 29:753,
 29:763, 33:842, 33:1209, 50:1011,
 50:1077, 52:143, 56:8, 57:275, 1205
\inputlineno 14:214, 01:311, 1207
\insert 45:465,
 45:484, 45:502, 54:521, 54:522,
 54:1887, 02:206, 02:254, 02:279,
 02:281, 02:284, 02:299, 02:332, 402
\InsertMark 48:363, 48:483, 222
insertmark 48:249, 943
\int 30:355
int commands:
 \int_add:Nn 07:607, 07:1252
 \int_case:nnTF 55:19, 55:29
 \int_compare:nNnTF
 08:2374, 08:2571, 09:164, 09:239,
 09:375, 09:487, 10:39, 11:45, 28:287,
 28:326, 48:79, 48:254, 53:55, 53:111,
 53:142, 53:353, 53:359, 53:367,
 55:136, 55:145, 55:168, 55:187,
 55:190, 55:215, 55:222, 55:233,
 07:457, 07:602, 07:873, 07:915,
 07:1934, 07:1987, 07:2255, 07:2257,
 08:1583, 08:1605, 08:1616, 08:1666,
 08:1694, 08:1705, 08:2065, 08:2071
 \int_compare:nTF 11:191
 \int_compare_p:nNn .. 08:2918, 08:2919
 \int_const:Nn 10:37
 \int_decr:N 07:536,
 07:582, 07:1933, 08:1594, 08:1680
 \int_div_truncate:nn 07:1401
 \int_eval:n 11:362, 05:163,
 53:355, 07:2269, 08:1248, 08:1602,
 08:1691, 08:1747, 08:1754, 08:1852
 \int_gadd:Nn 55:125, 55:228
 \int_gdecr:N 52:559, 55:199
 \int_gincr:N 17:37, 48:255, 52:549,
 53:87, 53:102, 55:179, 55:196, 07:1075
 \int_gset:Nn 28:292, 28:331, 52:545,
 55:138, 55:183, 55:205, 55:206, 55:225
 \int_gzero:N 48:256,
 55:174, 55:175, 55:180, 55:193, 55:202
 \int_if_odd:nTF 48:403, 48:462
 \int_if_zero:nTF 10:33
 \int_incr:N 07:360, 07:480,
 07:887, 07:957, 07:964, 07:1219,
 07:1519, 07:1943, 08:1562, 08:1644, 149
 \int_new:N 09:7, 11:34, 17:23,
 48:249, 52:544, 53:346, 53:348,
 55:38, 55:100, 55:101, 55:116,
 07:15, 07:27, 07:36, 07:1913, 08:1549
\int_set:Nn . 09:159, 09:202, 09:234,
 09:277, 09:447, 09:448, 09:478,
 11:190, 52:212, 52:226, 07:1097,
 07:1108, 07:1316, 07:1326, 07:1444,
 07:1917, 07:2415, 07:2441, 08:1164
\int_set_eq:NN 48:68
\int_step_function:nN 07:1399
\int_step_inline:nnn
 09:167, 09:181, 09:242, 09:256
\int_use:N 09:220,
 09:523, 09:530, 10:58, 10:84, 10:103,
 17:38, 28:281, 28:282, 28:284,
 28:287, 28:314, 28:315, 28:317,
 28:326, 36:233, 36:243, 36:244,
 48:104, 48:106, 48:225, 52:553,
 53:104, 53:116, 53:363, 53:371,
 55:20, 55:30, 55:149, 55:159, 55:160,
 55:211, 55:212, 07:25, 07:918,
 07:1079, 07:1108, 07:1237, 07:1241,
 07:1242, 07:1326, 07:1510, 1064
\int_value:w 53:48, 53:137
\int_zero:N 55:45,
 07:351, 07:445, 07:782, 07:786,
 07:922, 07:1211, 08:1558, 08:1640
\c_max_int
 48:68, 53:214, 53:240, 53:265, 53:292
\l_tmpa_int 346
\c_zero_int 09:164, 09:239, 53:91
int internal commands:
 \g__mark_int
 48:225, 48:249, 48:254, 48:255, 48:256
\interdisplaylinepenalty
 18:14, 38:55, 38:207, 38:393
\interfootlinepenalty 02:427
\interfootnotelinepenalty
 18:19, 45:467, 45:486, 45:504, 02:427
\interlinepenalty 18:11, 24:689, 37:536,
 37:539, 37:557, 37:560, 44:67,
 44:118, 44:209, 44:232, 45:467,
 45:486, 45:504, 54:340, 54:1157,
 54:1161, 54:1323, 54:1327, 02:331, 1195
\inteval 77
\inteval 05:158, 05:173, 77
\intextsep . 54:1140, 54:1144, 54:1159,
 54:1162, 54:1169, 54:1302, 54:1308,
 54:1325, 54:1328, 54:1337, 54:2341
\intop 30:354, 30:355
\iota 30:283
iow commands:
 \iow_char:N
 08:2594, 08:2596, 08:2598,
 08:2603, 08:2606, 08:2608, 08:2618,
 08:2619, 08:2659, 08:2667, 08:2679,
 08:2684, 08:2689, 08:2875, 08:2877,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

09:581, 16:153, 16:160, 16:161,  
 16:162, 16:163, 07:468, 07:532,  
 07:563, 07:635, 07:704, 07:716,  
 07:747, 07:1118, 07:1124, 07:1918,  
 07:2722, 07:2800, 08:1777, 08:1788  
`\iow_log:n` ..... 08:1820  
`\iow_newline:` .....  
     ..... 52:557, 07:1369, 07:1389,  
     07:1398, 07:1416, 07:1432, 07:1517  
`\iow_now:Nn` ..... 53:362  
`\iow_term:n` ..... 08:2108, 08:2113,  
     09:19, 09:28, 09:39, 09:48, 09:50,  
     09:66, 09:70, 10:25, 48:20, 48:90,  
     48:164, 48:170, 48:176, 48:229,  
     52:550, 52:583, 07:1432, 08:531,  
     08:604, 08:1436, 08:1439, 08:1457,  
     08:1460, 08:1478, 08:1512, 08:1598,  
     08:1618, 08:1619, 08:1621, 08:1685,  
     08:1707, 08:1708, 08:1710, 08:1776,  
     08:1787, 08:1801, 08:1807, 08:1808,  
     08:1809, 08:1812, 08:1816, 08:1825  
`\ishortstack` ..... 42:132  
`\itdefault` ..... 25:697, 29:30, 30:102  
`\item` ... 14:282, 37:444, 37:494, 37:496,  
     37:527, 37:549, 38:469, 38:481,  
     38:508, 39:165, 39:243, 41:78, 43:53,  
     43:55, 43:60, 43:62, 47:4, 47:8, 1202  
`\itemindent` .....  
     39:9, 39:42, 39:95, 39:211, 39:232, 803  
`\itemitem` ..... 1195  
`\itemize` (env.) ..... 39:266  
`\itemize` ..... 39:266  
`\itemsep` ..... 39:1, 39:200, 803  
`\iterate` ..... 01:65, 01:66, 02:490  
`\itshape` ..... 21:447, 21:807,  
     25:695, 25:696, 29:28, 29:29, 29:554,  
     29:587, 29:593, 32:21, 33:634, 43:53,  
     43:55, 43:60, 43:62, 45:399, 1240

**J**

`\J` ..... 57:257, 57:552  
`\j` ..... 21:248, 21:403, 21:548,  
     21:779, 21:1150, 21:1365, 21:1451  
`\jmath` ..... 30:317  
`\jobname` ..... 1046  
`\Join` ..... 29:725  
`\joinrel` ..... 30:478, 30:485,  
     30:487, 30:489, 30:491, 30:493,  
     30:495, 30:497, 30:499, 30:503, 30:505  
`\jot` ..... 38:53, 38:204, 38:404, 38:414

**K**

`\k` ..... 21:485, 21:590, 21:595,  
     21:617, 21:622, 21:698, 21:699,

`\kanjiskip` ..... 05:74  
`\kappa` ..... 30:284  
`\ker` ..... 38:27  
`\kern` ..... 1076  
`\kern` ..... 1233  
`kernel` (`refstepcounter/target`) (`plug`)  
     ..... 35:119  
`kernel` (`tagsupport/recordtarget`)  
     (`plug`) ..... 55:50  
`kernel` internal commands:  
     `\_kernel_chk_if_free_cs:N` .....  
         ..... 07:3006, 193  
     `\_kernel_cmd_if_xparse:NTF` .....  
         ..... 09:128, 07:1128,  
         07:1439, 07:1441, 07:2473, 07:2494, 144  
     `\_kernel_cs_parameter_spec:N` ...  
         ..... 09:161,  
         09:236, 09:369, 05:142, 07:2477, 1268  
     `\_kernel_cs_parm_from_arg_-`  
         `count:nnTF` ..... 08:109  
     `\_kernel_exp_not:w` .....  
         ..... 08:48, 08:50, 08:56, 08:61  
     `\l\_kernel_expl_bool` ..... 09:380  
     `\_kernel_file_name_sanitize:n` 52:98  
     `\_kernel_msg` ..... 1257  
     `\_kernel_msg` ..... 1259  
     `\_kernel_quark_new_conditional:Nn` .....  
         ..... 11:41  
     `\_kernel_quark_new_test:N` ... 07:48  
     `\g\_kernel_target_int` .....  
         ..... 17:23, 17:37, 17:38  
`\kerneltmpDoNotUse` ... 09:440, 09:452,  
     09:458, 09:463, 09:511, 09:518,  
     09:539, 09:547, 09:554, 09:567, 323

keys commands:

`\keys_define:nn` .. 11:505, 11:526,  
     11:587, 20:744, 51:70, 51:79, 51:177,  
     51:185, 51:189, 51:206, 51:241, 57:572  
`\keys_if_exist:nnTF` .....  
     ..... 51:68, 51:118, 51:124, 51:148  
`\l\_keys_key_str` .....  
     ..... 51:48, 51:49, 51:75, 51:194  
`\l\_keys_key_tt` ..... 51:48  
`\l\_keys_path_str` ..... 51:47  
`\keys_set:nn` .... 51:83, 51:256, 57:590  
`\keys_set_known:nnN` ..... 11:1195  
`\l\_keys_usage_load_prop` ..... 51:202  
`\l\_keys_usage_preamble_prop` . 51:237

**File Key:** 01=`ltdirchk.dtx`, 02=`ltpplain.dtx`, 03=`ltvers.dtx`, 04=`ltluatex.dtx`, 05=`ltexpl.dtx`,  
 06=`ltdefns.dtx`, 07=`ltcmd.dtx`, 08=`lthooks.dtx`, 09=`ltcmhooks.dtx`, 10=`ltsockets.dtx`,  
 11=`lttemplates.dtx`, 12=`ltalloc.dtx`, 13=`ltcntrl.dtx`, 14=`lterror.dtx`, 15=`ltpar.dtx`,  
 16=`ltpara.dtx`, 17=`ltmeta.dtx`, 18=`ltspaced.dtx`, 19=`ltlogos.dtx`, 20=`ltfiles.dtx`,  
 21=`ltoutenc.dtx`, 22=`ltcounts.dtx`, 23=`ltlength.dtx`, 24=`ltfssbas.dtx`, 25=`ltfssaxes.dtx`,  
 26=`ltfsstrc.dtx`, 27=`ltfsscmp.dtx`, 28=`ltfssdcl.dtx`, 29=`ltfssini.dtx`, 30=`fontdef.dtx`,  
 31=`preload.dtx`, 32=`ltfntcmd.dtx`, 33=`lttextcomp.dtx`, 34=`ltpageno.dtx`, 35=`ltxref.dtx`,  
 36=`ltproperties.dtx`, 37=`ltmisen.dtx`, 38=`ltmath.dtx`, 39=`ltlists.dtx`, 40=`ltboxes.dtx`,  
 41=`lttab.dtx`, 42=`ltpictur.dtx`, 43=`ltthm.dtx`, 44=`ltsect.dtx`, 45=`ltfloat.dtx`,  
 46=`ltdxglo.dtx`, 47=`ltbibl.dtx`, 48=`ltmarks.dtx`, 49=`ltpage.dtx`, 50=`ltclass.dtx`,  
 51=`ltkeys.dtx`, 52=`ltfilehook.dtx`, 53=`ltshipout.dtx`, 54=`ltoutput.dtx`, 55=`lttagging.dtx`,  
 56=`lthyphen.dtx`, 57=`ltfinal.dtx`

keys internal commands:

- \l\_\_keys\_class\_only\_clist . . . . .
- . . . . . 51:54, 51:121, 51:150
- \l\_\_keys\_find\_key\_module:wNN . . . . . 51:46
- \l\_\_keys\_forced\_global\_clist . . . . .
- . . . . . 51:49, 51:55, 51:152
- \l\_\_keys\_no\_value\_bool . . . . . 51:24
- \l\_\_keys\_option\_end: . . . . . 51:64, 51:78, 51:86
- \l\_\_keys\_options:n . . . . . 51:58, 51:58, 51:198
- \l\_\_keys\_options\_aux:n . . . . .
- . . . . . 51:58, 51:59, 51:60
- \l\_\_keys\_options\_class:n . . . . .
- . . . . . 51:100, 51:104, 51:104
- \l\_\_keys\_options\_class:nn . . . . .
- . . . . . 51:104, 51:120, 51:125, 51:132
- \l\_\_keys\_options\_class:nnn . . . . .
- . . . . . 51:104, 51:110, 51:116
- \l\_\_keys\_options\_clist . . . . .
- . . . . . 51:56, 51:65,
- . . . . . 51:82, 51:135, 51:160, 51:169, 1030
- \l\_\_keys\_options\_end: . . . . . 51:58
- \l\_\_keys\_options\_expand\_module:Nn . . . . .
- . . . . . 51:59,
- . . . . . 51:178, 51:178, 51:185, 51:189, 51:256
- \l\_\_keys\_options\_expand\_module:nN . . . . .
- . . . . . 51:178, 51:182
- \l\_\_keys\_options\_global:n . . . . .
- . . . . . 51:66, 51:95, 51:95
- \l\_\_keys\_options\_loaded:n . . . . .
- . . . . . 51:87, 51:200, 51:200
- \l\_\_keys\_options\_loaded:nn . . . . .
- . . . . . 51:200, 51:209, 51:214
- \l\_\_keys\_options\_loading\_bool . . . . .
- . . . . . 51:57, 51:81, 51:84, 51:216, 1029
- \l\_\_keys\_options\_local: . . . . .
- . . . . . 51:67, 51:163, 51:163
- \l\_\_keys\_options\_package:n . . . . .
- . . . . . 51:101, 51:137, 51:137
- \l\_\_keys\_options\_package:nn . . . . .
- . . . . . 51:137, 51:153, 51:155, 51:158
- \l\_\_keys\_options\_package:nnn . . . . .
- . . . . . 51:137, 51:141, 51:146
- \c\_\_keys\_props\_root\_str . . . . .
- . . . . . 51:9, 51:10, 51:22
- \l\_\_keys\_remove\_equals:n . . . . .
- . . . . . 51:111, 51:142, 51:174, 51:174
- \l\_\_keys\_remove\_equals:w . . . . .
- . . . . . 51:174, 51:175, 51:176
- \l\_\_keys\_scope:N . . . . .
- . . . . . 51:22, 51:33, 51:35, 51:44
- \l\_\_keys\_scope:n . . . . .
- . . . . . 51:22, 51:25, 51:26, 51:28
- \l\_\_keys\_tmp:nn . . . . . 51:5, 51:11, 51:13
- \l\_\_keys\_tmpa\_tl . . . . . 51:202, 51:204

keyval commands:

- \keyval\_parse>NNn . . . . . 11:212, 11:610
- \keyval\_parse:nnn . . . . . 11:406, 11:527
- \KeyValue . . . . .
- . . . . . 11:79, 11:874, 11:1079, 11:1192, 345
- \kill . . . . . 41:154, 41:162

**L**

- \L . . . . . 21:242, 21:424,
- . . . . . 21:530, 21:771, 21:1143, 50:1332,
- . . . . . 50:1464, 50:1554, 50:1576, 57:650
- \l . . . . . 21:251,
- . . . . . 21:426, 21:551, 21:780, 21:1144, 57:650
- \label . . . . . 35:94, 35:160,
- . . . . . 35:164, 44:182, 48:251, 49:25, 49:36,
- . . . . . 49:53, 49:61, 54:623, 54:682, 1235
- \label . . . . . 36:236, 750
- \labelenumi . . . . . 816
- \labelenumiv . . . . . 816
- \labelformat . . . . . 739
- \labelformat . . . . . 35:150,
- . . . . . 35:177, 35:182, 35:188, 35:193, 35:199
- \labelitemi . . . . . 816
- \labelitemii . . . . . 816
- \labelitemiii . . . . . 816
- \labelitemiv . . . . . 816
- \labelsep . . . . . 39:9, 39:234,
- . . . . . 39:240, 43:53, 43:55, 43:60, 43:62, 803
- \labelwidth . . . . .
- . . . . . 39:9, 39:93, 39:233, 39:235, 39:238, 803
- \Lambda . . . . . 30:307
- \lambda . . . . . 30:285
- \land . . . . . 30:375, 30:377
- \langle . . . . . 30:601
- \language . . . . . 20:52,
- . . . . . 20:122, 37:532, 37:724, 54:601,
- . . . . . 02:35, 56:10, 02:82, 02:84, 02:99, 1246
- \lastbox . . . . . 24:707,
- . . . . . 38:193, 38:194, 39:132, 39:138,
- . . . . . 39:209, 44:99, 44:132, 54:307, 409
- \LastDeclaredEncoding . . . . .
- . . . . . 24:137, 24:140, 57:482, 1237
- \LastMark . . . . . 48:366, 48:485, 944
- \lastnamedcs . . . . . 06:758
- \lastnodetype 24:700, 24:701, 24:702, 24:706
- \lastpenalty . . . . . 24:703, 32:112, 32:115
- \lastskip . . . . . 18:45, 18:129,
- . . . . . 18:141, 18:160, 18:221, 18:222,
- . . . . . 18:226, 18:228, 18:229, 18:241,
- . . . . . 18:256, 18:273, 18:295, 18:298,
- . . . . . 18:328, 18:331, 18:362, 18:365,
- . . . . . 18:366, 32:102, 32:105, 39:115,
- . . . . . 39:116, 39:174, 39:175, 42:123,
- . . . . . 02:514, 02:515, 02:517, 02:519, 1196

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\LaTeX ... [19:3](#), [19:15](#), [50:1294](#), [50:1427](#),  
[50:1516](#), [53:397](#), [53:403](#), [06:837](#), [02:381](#)  
\LaTeXe ... [19:13](#)  
\latexrelease ... [1057](#)  
\LaTeXReleaseInfo ... [03:36](#), [03:37](#),  
[03:40](#), [03:45](#), [03:50](#), [37:47](#), [37:112](#), [38](#)  
\latexreleaseversion ... [03:1](#)  
\lbrace ... [21:308](#), [30:605](#)  
\lbrack ... [02:441](#)  
\lccode ... [14:19](#), [14:20](#),  
[14:21](#), [14:22](#), [14:23](#), [14:24](#), [21:140](#),  
[21:1065](#), [37:653](#), [37:670](#), [37:684](#),  
[37:699](#), [37:742](#), [57:224](#), [57:241](#),  
[57:249](#), [57:256](#), [57:258](#), [57:259](#),  
[57:261](#), [57:263](#), [57:264](#), [57:265](#),  
[57:266](#), [57:536](#), [57:544](#), [57:551](#),  
[57:553](#), [57:554](#), [57:556](#), [57:558](#), [1243](#)  
\lceil ... [30:609](#)  
\lcode ... [1234](#)  
\ldotp ... [30:508](#), [30:511](#), [30:626](#)  
\ldots ... [21:322](#), [30:512](#), [1222](#)  
\le ... [30:423](#), [30:425](#)  
\leaders ... [30:347](#), [30:565](#), [30:566](#), [30:568](#),  
[30:569](#), [41:376](#), [42:564](#), [42:577](#),  
[42:589](#), [42:599](#), [44:214](#), [44:237](#), [02:546](#)  
\leadsto ... [29:728](#)  
\leavevmode ... [18:454](#), [18:468](#),  
[21:75](#), [21:184](#), [21:289](#), [21:290](#),  
[21:393](#), [21:425](#), [21:429](#), [21:432](#),  
[21:479](#), [21:763](#), [21:796](#), [32:123](#),  
[33:98](#), [33:924](#), [37:536](#), [37:557](#),  
[37:570](#), [37:581](#), [37:611](#), [37:625](#),  
[37:720](#), [37:730](#), [37:743](#), [38:469](#),  
[38:481](#), [38:508](#), [39:58](#), [39:103](#),  
[40:8](#), [40:17](#), [40:24](#), [40:195](#), [40:197](#),  
[40:213](#), [40:245](#), [40:264](#), [40:331](#),  
[40:358](#), [40:437](#), [40:544](#), [40:561](#),  
[40:568](#), [41:178](#), [42:134](#), [42:314](#),  
[42:373](#), [44:40](#), [44:210](#), [44:222](#),  
[44:233](#), [45:530](#), [47:34](#), [54:159](#),  
[54:164](#), [54:186](#), [54:191](#), [02:505](#),  
[02:532](#), [02:535](#), [02:546](#), [02:548](#), [1224](#)  
\left ... [30:632](#), [30:634](#),  
[30:636](#), [30:638](#), [30:643](#), [30:644](#),  
[30:645](#), [30:646](#), [38:167](#), [38:173](#), [38:195](#)  
\Leftarrow ... [30:417](#), [30:499](#), [30:505](#)  
\leftarrow ... [30:444](#),  
[30:446](#), [30:487](#), [30:497](#), [30:503](#), [30:557](#)  
\leftarrowfill ... [30:541](#), [30:557](#)  
\lefteqn ... [38:428](#)  
\leftharpoondown ... [30:460](#), [30:474](#)  
\leftharpoonup ... [30:459](#)  
\lefthyphenmin ... [56:11](#), [02:371](#)  
\leftline ... [40:614](#)

\leftmargin ... [39:9](#),  
[39:52](#), [39:53](#), [39:94](#), [39:170](#), [39:172](#), [803](#)  
\leftmargini ... [38:461](#), [39:17](#), [803](#)  
\leftmarginii ... [39:17](#)  
\leftmarginiii ... [39:17](#)  
\leftmarginiv ... [39:17](#)  
\leftmarginv ... [39:17](#)  
\leftmarginvi ... [39:17](#), [803](#)  
\leftmark ... [49:77](#), [947](#)  
\Leftarrow ... [30:416](#)  
\rightarrow ... [30:443](#)  
\leftskip ... [24:684](#),  
[37:452](#), [37:458](#), [37:462](#), [37:472](#),  
[37:476](#), [37:480](#), [37:529](#), [37:551](#),  
[39:74](#), [40:392](#), [40:413](#), [44:207](#),  
[44:212](#), [44:230](#), [44:235](#), [02:416](#), [02:527](#)

legacy commands:

\legacy\_if:nTF ... [48:401](#), [48:422](#), [48:460](#), [48:467](#)  
\legacyoldstylenums ... [33:4](#), [33:617](#)  
\leq ... [30:421](#), [30:423](#)  
\leqno ... [38:439](#)  
\let ... [1213](#)  
\LetLtxMacro ... [101](#)  
\lfloor ... [30:613](#)  
\lg ... [38:4](#)  
\lgrouop ... [30:615](#)  
\lhd ... [29:731](#)  
\lhook ... [30:484](#), [30:485](#)  
\lim ... [38:6](#)  
\liminf ... [38:8](#)  
\limits ... [30:546](#), [30:550](#), [38:162](#), [38:353](#)  
\limsup ... [38:7](#)  
\line ... [14:269](#), [42:158](#), [42:450](#), [42:808](#), [42:825](#)  
\linebreak ... [421](#)  
\linebreak ... [18:9](#), [18:27](#)  
\linepenalty ... [02:320](#)  
\lineskip ... [30:465](#), [38:200](#), [40:394](#), [40:414](#),  
[41:71](#), [41:198](#), [42:136](#), [42:315](#),  
[42:374](#), [53:223](#), [53:274](#), [54:626](#),  
[54:685](#), [02:410](#), [02:435](#), [02:500](#), [02:535](#)  
\lineskiplimit ... [30:465](#), [30:517](#), [38:202](#), [38:206](#),  
[40:380](#), [40:395](#), [40:402](#), [53:224](#),  
[53:275](#), [54:626](#), [54:685](#), [02:396](#),  
[02:436](#), [02:500](#), [02:537](#), [02:538](#), [1246](#)  
\linespread ... [24:316](#)  
\linethickness ... [42:130](#), [42:809](#), [42:826](#)  
\linewidth ... [20:30](#), [20:100](#),  
[20:158](#), [38:298](#), [38:324](#), [38:470](#),  
[38:482](#), [38:509](#), [38:513](#), [38:532](#), [39:9](#),  
[39:51](#), [39:52](#), [39:54](#), [40:390](#), [40:411](#),  
[41:36](#), [45:266](#), [54:148](#), [54:207](#), [802](#)  
\LinkTargetOff ... [17:20](#)

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltphypen.dtx, 57=ltfinal.dtx

|                                                         |                                                                                          |                                                                                                    |
|---------------------------------------------------------|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| \LinkTargetOn . . . . .                                 | <u>17:20</u>                                                                             | lua commands:                                                                                      |
| list (env.) . . . . .                                   | <u>39:34</u>                                                                             | \lua_now:n . . . . . <u>53:31</u>                                                                  |
| \list . . . . .                                         | <u>39:34</u> , <u>39:260</u> , <u>39:271</u>                                             | \luabytecode . . . . . <u>04:202</u>                                                               |
| \listfiles . . . . .                                    | <u>978</u>                                                                               | \luachunk . . . . . <u>04:210</u>                                                                  |
| \listfiles . . . . .                                    | <u>20:750</u> , <u>220</u>                                                               | \luadef . . . . . <u>04:182</u> , <u>04:186</u> , <u>42</u>                                        |
| \listparindent . . . . .                                | <u>39:9</u> , <u>39:41</u> , <u>39:50</u> , <u>803</u>                                   | \luafunction . . . <u>04:178</u> , <u>04:182</u> , <u>04:186</u> , <u>42</u>                       |
| \literal . . . . .                                      | <u>1202</u>                                                                              | \luatexbase . . . . . <u>04:288</u>                                                                |
| \ll . . . . .                                           | <u>30:441</u>                                                                            | \luatexluafunction . . . . . <u>01:21</u> , <u>01:26</u>                                           |
| \llap . . . . .                                         | <u>39:262</u> , <u>39:273</u> , <u>40:618</u>                                            | \luatexversion . . . . .                                                                           |
| \lmoustache . . . . .                                   | <u>30:570</u>                                                                            | <u>04:5</u> , <u>21:994</u> , <u>01:14</u> , <u>05:73</u> , <u>05:119</u>                          |
| \ln . . . . .                                           | <u>38:5</u>                                                                              |                                                                                                    |
| \lnot . . . . .                                         | <u>30:333</u> , <u>30:334</u>                                                            |                                                                                                    |
| \LoadClass . . . . .                                    | <u>976</u>                                                                               | <b>M</b>                                                                                           |
| \LoadClass . . . . .                                    | <u>50:686</u> , <u>50:716</u> , <u>50:959</u> , <u>50:1094</u> ,                         | \M . . . . . <u>02:437</u>                                                                         |
|                                                         | <u>50:1189</u> , <u>50:1197</u> , <u>50:1198</u> , <u>1198</u>                           | \Macro . . . . . <u>1195</u>                                                                       |
| \LoadClassWithOptions . . . . .                         | <u>976</u>                                                                               | \mag . . . . . <u>02:376</u>                                                                       |
| \LoadClassWithOptions . . . . .                         | <u>50:715</u>                                                                            | \magstep . . . . . <u>02:428</u>                                                                   |
| \LoadFontDefinitionFile <u>24:443</u> , <u>24:469</u> , | <u>24:470</u> , <u>30:21</u> , <u>30:27</u> , <u>30:28</u> , <u>30:29</u> , <u>30:33</u> | \magstephalf . . . . . <u>02:428</u>                                                               |
| \LoadPackageWithOptions . . . . .                       | <u>1037</u>                                                                              | \makeat... . . . . . <u>1214</u>                                                                   |
| \loccount . . . . .                                     | <u>04:17</u>                                                                             | \makeatletter . . . . . <u>09:376</u> ,                                                            |
| \log . . . . .                                          | <u>38:3</u>                                                                              | <u>20:32</u> , <u>20:102</u> , <u>20:160</u> , <u>24:421</u> , <u>37:27</u> ,                      |
| \loggingall . . . . .                                   | <u>02:557</u>                                                                            | <u>37:92</u> , <u>37:154</u> , <u>44:151</u> , <u>50:680</u> ,                                     |
| \loggingoutput . . . . .                                | <u>02:571</u> ,                                                                          | <u>50:923</u> , <u>50:1055</u> , <u>54:2</u> , <u>06:833</u> , <u>1236</u>                         |
|                                                         | <u>02:589</u> , <u>02:605</u> , <u>02:619</u> , <u>02:553</u> , <u>1239</u>              |                                                                                                    |
| \LogHook . . . . .                                      | <u>08:2796</u> , <u>205</u>                                                              | \makeatother . . . . . <u>09:376</u> , <u>50:680</u> , <u>06:833</u> , <u>57:712</u> , <u>1211</u> |
| \LogSocket . . . . .                                    | <u>10:170</u> , <u>10:184</u> , <u>335</u>                                               | \makebox . . . . . <u>818</u>                                                                      |
| \long . . . . .                                         | <u>1253</u>                                                                              | \makebox . . . . . <u>38:298</u> , <u>38:324</u> , <u>40:3</u>                                     |
| \Longleftarrow . . . . .                                | <u>30:499</u>                                                                            | \makeglossary . . . . . <u>934</u>                                                                 |
| \longleftarrow . . . . .                                | <u>30:496</u>                                                                            | \makeglossary . . . . . <u>20:205</u> , <u>46:20</u> , <u>1219</u>                                 |
| \Longleftrightarrow . . . . .                           | <u>30:505</u> , <u>30:507</u>                                                            | \makeindex . . . . . <u>934</u>                                                                    |
| \longleftrightarrow . . . . .                           | <u>30:503</u>                                                                            | \makeindex . . . . . <u>20:204</u> , <u>46:3</u> , <u>1219</u>                                     |
| \longmapsto . . . . .                                   | <u>30:501</u>                                                                            | \makelabel . . . . . <u>39:45</u> , <u>39:97</u> ,                                                 |
| \Longrightarrow . . . . .                               | <u>30:493</u>                                                                            | <u>39:229</u> , <u>39:242</u> , <u>39:262</u> , <u>39:273</u> , <u>802</u>                         |
| \longrightarrow . . . . .                               | <u>30:494</u> , <u>30:501</u>                                                            | \MakeLinkTarget . . . <u>17:20</u> , <u>43:53</u> , <u>43:55</u> , <u>420</u>                      |
| \loop . . . . .                                         | <u>04:150</u> , <u>04:159</u> , <u>24:698</u> ,                                          | \MakeLowercase . . . . . <u>57:571</u> , <u>1226</u>                                               |
|                                                         | <u>41:382</u> , <u>50:1242</u> , <u>50:1303</u> , <u>50:1373</u> ,                       | \MakeRobust . . . <u>28:888</u> , <u>28:1083</u> , <u>42:805</u> ,                                 |
|                                                         | <u>50:1436</u> , <u>50:1487</u> , <u>50:1525</u> , <u>57:365</u> ,                       | <u>42:806</u> , <u>42:807</u> , <u>42:808</u> , <u>42:809</u> ,                                    |
|                                                         | <u>57:376</u> , <u>57:386</u> , <u>57:397</u> , <u>57:427</u> ,                          | <u>42:810</u> , <u>42:811</u> , <u>42:812</u> , <u>42:813</u> ,                                    |
|                                                         | <u>57:453</u> , <u>57:463</u> , <u>01:65</u> , <u>02:490</u> , <u>1220</u>               | <u>42:814</u> , <u>42:815</u> , <u>42:816</u> , <u>06:283</u> ,                                    |
| \looseness . . . . .                                    | <u>02:337</u>                                                                            | <u>06:878</u> , <u>06:879</u> , <u>06:880</u> , <u>06:881</u> ,                                    |
| \lor . . . . .                                          | <u>30:376</u> , <u>30:378</u>                                                            | <u>06:882</u> , <u>06:883</u> , <u>06:884</u> , <u>06:885</u> ,                                    |
| \lower . . . . .                                        | <u>19:2</u> , <u>30:465</u> ,                                                            | <u>06:886</u> , <u>06:887</u> , <u>06:888</u> , <u>06:889</u> ,                                    |
|                                                         | <u>40:278</u> , <u>42:35</u> , <u>42:45</u> , <u>42:196</u> , <u>42:305</u> ,            | <u>06:890</u> , <u>06:891</u> , <u>06:892</u> , <u>06:893</u> , <u>1254</u>                        |
|                                                         | <u>42:306</u> , <u>42:353</u> , <u>42:354</u> , <u>42:409</u> , <u>42:410</u>            |                                                                                                    |
| \lowercase . . . . .                                    | <u>14:26</u> , <u>21:141</u> , <u>21:1066</u> ,                                          | \maketitle . . . . . <u>898</u>                                                                    |
|                                                         | <u>21:1544</u> , <u>24:332</u> , <u>24:415</u> , <u>37:657</u> ,                         | \MakeTitlecase . . . . . <u>57:571</u> , <u>1262</u>                                               |
|                                                         | <u>37:674</u> , <u>37:688</u> , <u>37:703</u> , <u>37:743</u> , <u>1232</u>              | \MakeUppercase . . . . . <u>35:152</u> ,                                                           |
| \lq . . . . .                                           | <u>02:439</u>                                                                            | <u>35:154</u> , <u>35:179</u> , <u>35:190</u> , <u>57:571</u> , <u>1226</u>                        |
| \lrbox (env.) . . . . .                                 | <u>818</u>                                                                               | \mapsto . . . . . <u>30:451</u>                                                                    |
| \lrbox . . . . .                                        | <u>40:183</u>                                                                            | \mapstochar . . . . . <u>30:450</u> , <u>30:451</u> , <u>30:501</u>                                |
| \ltfilehookdate . . . . .                               | <u>52:542</u>                                                                            | \margininpar . . . . . <u>45:308</u> , <u>404</u>                                                  |
| \ltfilehookversion . . . . .                            | <u>52:542</u>                                                                            | \margininparpush . . . . . <u>54:85</u> , <u>54:1838</u>                                           |
|                                                         |                                                                                          | \margininparsep . . . . . <u>54:84</u> , <u>54:1849</u> , <u>54:1851</u>                           |
|                                                         |                                                                                          | \margininparwidth . . . . . <u>45:341</u> , <u>45:359</u> , <u>54:83</u> , <u>54:1851</u>          |
|                                                         |                                                                                          | \mark . . <u>49:31</u> , <u>49:42</u> , <u>49:56</u> , <u>49:64</u> , <u>49:82</u> , <u>942</u>    |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=ltphypen.dtx, 57=ltfinal.dtx

mark commands:

- \mark\_debug\_off: . . . . .  
..... 48:305, 48:311, 48:322, 946
- \mark\_debug\_on: . . . . .  
..... 48:305, 48:306, 48:321, 946
- \mark\_if\_eq:nmmn . . . . . 48:265
- \mark\_if\_eq:nnnnnn . . . . . 48:272
- \mark\_if\_eq:nnnnnnTF . . . . . 48:265, 945
- \mark\_if\_eq:nnnnTF . . . . .  
... 48:265, 48:373, 48:376, 48:379, 945
- \mark\_insert:nn . . . . . 48:174, 48:183,  
48:216, 48:216, 48:365, 49:28,  
49:29, 49:30, 49:39, 49:40, 49:41, 957
- \mark\_new\_class:n . . . . .  
..... 48:7, 48:7, 48:16, 48:363, 943
- \mark\_use\_first:nn . . . . .  
..... 48:260, 48:260, 48:367, 944
- \mark\_use\_last:nn . . . . .  
..... 48:260, 48:261, 48:369, 944
- \mark\_use\_top:nn . . . . .  
..... 48:260, 48:262, 48:371, 944

mark internal commands:

- \\_mark\_class\_status:nn . . . . .  
..... 48:323, 48:324, 48:355
- \\_mark\_debug:n . . . . .  
.... 48:20, 48:90, 48:164, 48:170,  
48:176, 48:229, 48:305, 48:305,  
48:318, 48:360, 48:397, 48:456, 962
- \\_mark\_debug\_gset: . . . . .  
..... 48:305, 48:309, 48:314, 48:316
- \\_mark\_error:n . . . . .  
..... 48:205, 48:208, 48:209, 48:210, 48:213
- \\_mark\_extract\_and\_handle\_-  
marks:nn . . . . .  
.... 48:65, 48:65, 48:118, 48:150, 953
- \\_mark\_get\_from\_splitmarks: . . .  
..... 48:151, 48:156, 48:156
- \\_mark\_get\_marks\_for\_reinsertion:nNN  
..... 48:147, 48:147, 950
- \\_mark\_id:n . . . . . 48:42, 48:43,  
48:44, 48:45, 48:46, 48:47, 48:48,  
48:49, 48:50, 48:51, 48:52, 48:53,  
48:54, 48:55, 48:56, 48:57, 48:58,  
48:59, 48:225, 48:248, 48:248, 951
- \\_mark\_new\_class:nn 48:7, 48:14, 48:17
- \\_mark\_prepare\_and\_extract:nn . . .  
..... 48:70, 48:73, 48:73, 48:94, 953
- \\_mark\_status:n . . . . .  
..... 48:352, 48:352, 48:360, 48:399, 48:458
- \\_mark\_update\_dblcol\_structures:  
..... 48:412, 48:412, 48:478, 949
- \\_mark\_update\_singlecol\_-  
structures: . . . . .  
..... 48:381, 48:381, 48:476, 949

- \\_\\_mark\_update\_structure\_-  
alias:nn . . . . . 48:123, 48:194,  
48:194, 48:391, 48:392, 48:393,  
48:394, 48:424, 48:428, 48:429, 949
- \\_\\_mark\_update\_structure\_from\_-  
material:nn . . . . . 48:117, 48:117,  
48:384, 48:388, 48:415, 48:419, 949
- \\_\\_mark\_update\_structure\_from\_-  
splitmarks:n . . . . .  
..... 48:119, 48:122, 48:122, 955
- \\_\\_mark\_update\_structure\_to\_-  
err:n . . . . . 48:205, 48:205, 48:425, 949
- \\_\\_mark\_use:n . . . . .  
..... 48:260, 48:261, 48:262, 48:263, 48:264
- \\_\\_mark\_vbox\_set\_split\_to\_-  
maxdimen:NN 48:85, 48:99, 48:100, 954
- \markboth . . . . . 49:21,  
49:22, 49:49, 49:51, 49:70, 49:72, 942
- \markright . . . . . 49:22, 49:59, 49:73, 946
- \marks . . . . . 04:37, 57:10, 57:12
- math (env.) . . . . . 38:345
- \math . . . . . 38:345
- \mathaccemt . . . . . 1245
- \mathaccent . . . . . 28:805, 28:853, 28:887, 28:897
- \mathalpha . . . . .  
... 28:975, 28:1154, 30:176, 30:177,  
30:178, 30:179, 30:180, 30:181,  
30:182, 30:183, 30:184, 30:185,  
30:186, 30:187, 30:188, 30:189,  
30:190, 30:191, 30:192, 30:193,  
30:194, 30:195, 30:196, 30:197,  
30:198, 30:199, 30:200, 30:201,  
30:202, 30:203, 30:204, 30:205,  
30:206, 30:207, 30:208, 30:209,  
30:210, 30:211, 30:212, 30:213,  
30:214, 30:215, 30:216, 30:217,  
30:218, 30:219, 30:220, 30:221,  
30:222, 30:223, 30:224, 30:225,  
30:226, 30:227, 30:228, 30:229,  
30:230, 30:231, 30:232, 30:233,  
30:234, 30:235, 30:236, 30:237,  
30:304, 30:305, 30:306, 30:307,  
30:308, 30:309, 30:310, 30:311,  
30:312, 30:313, 30:314, 30:523,  
30:524, 30:525, 30:526, 30:527,  
30:528, 30:529, 30:530, 30:532, 30:535
- \mathbf . . . . . 29:14, 29:255,  
29:308, 29:353, 29:381, 29:475, 30:158
- \mathbin . . . . . 28:1159, 30:239, 30:240, 30:242,  
30:365, 30:366, 30:367, 30:368,  
30:371, 30:372, 30:373, 30:374,  
30:377, 30:378, 30:379, 30:380,  
30:381, 30:382, 30:383, 30:384,  
30:385, 30:386, 30:387, 30:388,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

30:389, 30:390, 30:391, 30:392,  
 30:393, 30:394, 30:395, 30:396,  
 30:397, 30:398, 30:399, 30:400,  
 30:401, 30:402, 30:403, 30:404, 38:37  
 $\backslash\mathcal$  ..... 30:157  
 $\backslash\mathchar$  ..... 28:916, 28:960,  
 30:342, 30:343, 30:624, 02:533, 1245  
 $\backslash\mathchardef$  12:3, 12:4, 12:5, 12:6, 21:70,  
 04:227, 28:951, 02:21, 02:22, 02:23,  
 02:24, 02:107, 02:110, 02:111, 1218  
 $\backslash\mathcharzero$  ..... 04:227  
 $\backslash\mathchoice$  ..... 38:61  
 $\backslash\mathclose$  ..... 28:1162, 30:238,  
 30:247, 30:249, 30:252, 30:257,  
 30:263, 30:265, 30:267, 30:573,  
 30:600, 30:604, 30:608, 30:612,  
 30:618, 38:43, 38:46, 38:49, 38:52  
 $\backslash\mathcode$  .. 28:948, 30:259, 30:260, 30:261  
 $\backslash\mathdefaultsmode$  ..... 38:436, 38:437  
 $\backslash\mathdollar$  ..... 21:307, 30:621, 1218  
 $\backslash\mathellipsis$  ..... 21:321, 30:626, 1218  
 $\backslash\mathfontset$  ..... 1193  
 $\backslash\mathgroup$  ..... 24:14, 26:303, 26:309,  
 26:315, 26:316, 26:327, 30:650, 33:8,  
 33:14, 33:614, 33:1136, 02:79, 1193  
 $\backslash\mathhexbox$  ..... 29:652, 02:533, 1225  
 $\backslash\mathindent$  ..... 38:459,  
 38:471, 38:483, 38:511, 38:522, 1204  
 $\backslash\mathinner$  . 30:511, 30:515, 30:520, 30:626  
 $\backslash\mathit$  25:696, 29:29, 30:160, 30:163, 30:624  
 $\backslash\mathnormal$  ..... 30:156  
 $\backslash\mathop$  28:1158, 30:348, 30:349, 30:350,  
 30:351, 30:352, 30:353, 30:354,  
 30:356, 30:357, 30:358, 30:359,  
 30:360, 30:361, 30:363, 30:364,  
 30:544, 30:547, 38:3, 38:4, 38:5,  
 38:6, 38:7, 38:8, 38:9, 38:10, 38:11,  
 38:12, 38:13, 38:14, 38:15, 38:16,  
 38:17, 38:18, 38:19, 38:20, 38:21,  
 38:22, 38:23, 38:24, 38:25, 38:26,  
 38:27, 38:28, 38:29, 38:30, 38:31,  
 38:32, 38:33, 38:34, 38:162, 38:353  
 $\backslash\mathopen$  28:1161, 30:248, 30:251, 30:256,  
 30:262, 30:264, 30:266, 30:571,  
 30:602, 30:606, 30:610, 30:614,  
 30:616, 38:41, 38:44, 38:47, 38:50  
 $\backslash\mathord$  ..... 28:975, 28:1157, 30:243,  
 30:250, 30:253, 30:258, 30:270,  
 30:271, 30:272, 30:274, 30:275,  
 30:276, 30:277, 30:278, 30:279,  
 30:280, 30:281, 30:282, 30:283,  
 30:284, 30:285, 30:286, 30:287,  
 30:288, 30:289, 30:290, 30:291,  
 30:292, 30:293, 30:294, 30:295,  
 30:296, 30:297, 30:298, 30:299,  
 30:300, 30:301, 30:302, 30:303,  
 30:315, 30:316, 30:317, 30:318,  
 30:319, 30:320, 30:321, 30:322,  
 30:323, 30:324, 30:325, 30:326,  
 30:327, 30:328, 30:329, 30:330,  
 30:331, 30:332, 30:334, 30:335,  
 30:336, 30:337, 30:338, 30:339,  
 30:340, 30:341, 30:531, 30:533,  
 30:534, 30:556, 30:557, 30:560,  
 30:561, 30:562, 30:563, 30:575,  
 30:577, 30:579, 30:582, 30:584,  
 30:598, 30:620, 30:621, 30:622, 30:623  
 $\backslash\mathpalette$  ..... 30:464, 30:468,  
 30:471, 38:60, 38:69, 38:99, 38:129  
 $\backslash\mathparagraph$  .....  
 . 21:310, 22:216, 22:228, 30:621, 1234  
 $\backslash\mathpunct$  ..... 28:1163,  
 30:241, 30:245, 30:508, 30:509, 30:510  
 $\backslash\mathrel$  ..... 28:1160, 30:244, 30:246,  
 30:254, 30:255, 30:268, 30:269,  
 30:345, 30:405, 30:406, 30:407,  
 30:408, 30:409, 30:410, 30:411,  
 30:412, 30:413, 30:414, 30:415,  
 30:416, 30:417, 30:418, 30:421,  
 30:422, 30:425, 30:426, 30:427,  
 30:428, 30:429, 30:430, 30:431,  
 30:432, 30:433, 30:434, 30:435,  
 30:436, 30:437, 30:439, 30:440,  
 30:441, 30:442, 30:443, 30:444,  
 30:445, 30:448, 30:449, 30:450,  
 30:452, 30:453, 30:454, 30:455,  
 30:456, 30:457, 30:458, 30:459,  
 30:460, 30:461, 30:462, 30:464,  
 30:468, 30:471, 30:478, 30:480,  
 30:483, 30:484, 30:486, 30:489,  
 30:491, 30:586, 30:588, 30:590,  
 30:592, 30:594, 30:596, 38:42,  
 38:45, 38:48, 38:51, 38:162, 38:353  
 $\backslash\mathring$  ..... 30:535, 1236  
 $\backslash\mathrm$  ..... 29:5,  
 29:406, 29:449, 29:481, 30:155, 1194  
 $\backslash\mathsection$  .....  
 . 21:311, 22:215, 22:227, 30:621, 1234  
 $\backslash\mathsf$  ..... 29:8,  
 29:411, 29:454, 29:484, 30:159, 30:162  
 $\backslash\mathsterling$  ..... 21:319, 30:621, 1218  
 $\backslash\mathstrut$  ... 38:84, 38:93, 38:171, 38:172  
 $\backslash\mathsurround$  ..... 02:400, 02:521, 1235  
 $\backslash\mathsymbol$  ..... 28:953  
 $\backslash\mathtt$  29:11, 29:416, 29:459, 29:487, 30:161  
 $\backslash\mathunderscore$  ..... 30:621, 1219  
 $\backslash\mathversion$  . 24:336, 29:614, 29:616, 1212  
 $\backslash\mathversion.$  ..... 1193

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\matrix . . . . . [38:169](#), [38:173](#), [38:180](#)  
 \max . . . . . [38:22](#)  
 \maxdeadcycles . . . . . [54:7](#), [02:373](#)  
 \maxdepth . . . . . [18:302](#),  
   [18:335](#), [20:60](#), [20:129](#), [20:184](#),  
   [54:92](#), [54:171](#), [54:172](#), [54:510](#),  
   [54:518](#), [54:550](#), [54:719](#), [54:728](#),  
   [54:768](#), [54:995](#), [57:152](#), [02:393](#), [1206](#)  
 \maxdimen . . . . . [02:569](#), [02:570](#),  
   [02:588](#), [02:604](#), [02:619](#), [24:670](#),  
   [24:680](#), [24:716](#), [24:731](#), [26:384](#),  
   [26:437](#), [30:465](#), [42:474](#), [42:502](#),  
   [42:532](#), [42:610](#), [42:627](#), [50:1597](#),  
   [50:1645](#), [50:1654](#), [53:351](#), [53:353](#),  
   [53:409](#), [54:293](#), [54:1857](#), [54:1877](#),  
   [54:1882](#), [54:2203](#), [54:2243](#), [54:2244](#),  
   [54:2246](#), [57:156](#), [02:309](#), [02:394](#),  
   [02:395](#), [02:500](#), [02:538](#), [02:554](#), [1195](#)  
 \mbox . . . . . [818](#)  
 \mbox . . . . . [19:13](#), [21:293](#), [21:409](#),  
   [21:568](#), [21:1165](#), [29:648](#), [30:513](#),  
   [40:11](#), [40:20](#), [40:24](#), [42:52](#), [45:409](#),  
   [45:416](#), [45:437](#), [45:444](#), [02:533](#), [1223](#)  
 \mddefault . . . . . [29:18](#), [29:284](#),  
   [29:290](#), [29:291](#), [29:292](#), [29:331](#),  
   [29:332](#), [29:333](#), [29:342](#), [29:369](#),  
   [29:385](#), [29:402](#), [29:443](#), [29:479](#),  
   [30:100](#), [30:112](#), [30:114](#), [30:128](#), [1259](#)  
 \mdseries . . . . . [29:16](#), [29:17](#),  
   [29:221](#), [29:277](#), [29:328](#), [29:329](#),  
   [29:363](#), [29:364](#), [29:383](#), [29:384](#),  
   [29:477](#), [29:478](#), [29:651](#), [32:20](#), [221](#)  
 mdseries . . . . . [29:420](#)  
 mdseries/defaults . . . . . [29:420](#)  
 \meaning . . . . . [28:627](#),  
   [28:640](#), [28:741](#), [28:806](#), [28:853](#),  
   [28:917](#), [28:1011](#), [28:1107](#), [28:1211](#),  
   [01:203](#), [01:212](#), [06:240](#), [06:302](#),  
   [06:340](#), [06:368](#), [06:451](#), [01:307](#), [06:804](#)  
 \medbreak . . . . . [06:885](#), [06:906](#), [02:515](#)  
 \mediumseries . . . . . [642](#)  
 \medmuskip . . . . . [30:652](#),  
   [38:36](#), [38:38](#), [38:224](#), [38:227](#), [38:241](#)  
 \medskip . . . . . [18:447](#), [02:518](#)  
 \medskipamount . . . . . [18:448](#), [18:450](#), [02:517](#)  
 \medspace . . . . . [38:214](#)  
 \MessageBreak [03:101](#), [14:3](#), [14:6](#), [14:13](#),  
   [14:33](#), [14:46](#), [14:60](#), [14:73](#), [14:220](#),  
   [14:222](#), [14:228](#), [14:235](#), [17:13](#),  
   [21:161](#), [21:988](#), [21:1549](#), [21:1551](#),  
   [21:1553](#), [21:1556](#), [24:34](#), [24:35](#),  
   [24:561](#), [24:595](#), [25:456](#), [26:20](#), [26:21](#),  
   [26:67](#), [26:88](#), [26:327](#), [26:478](#), [26:498](#),  
   [26:530](#), [26:546](#), [26:561](#), [26:574](#),  
   [27:31](#), [27:33](#), [28:280](#), [28:313](#), [28:582](#),  
   [28:591](#), [28:729](#), [29:61](#), [29:94](#), [32:144](#),  
   [33:23](#), [33:79](#), [33:81](#), [33:100](#), [33:848](#),  
   [33:850](#), [33:851](#), [33:852](#), [33:854](#),  
   [33:856](#), [33:857](#), [33:858](#), [33:859](#),  
   [33:860](#), [33:910](#), [33:912](#), [33:919](#),  
   [33:926](#), [33:1141](#), [05:78](#), [05:81](#),  
   [05:82](#), [05:83](#), [05:84](#), [05:85](#), [05:86](#),  
   [37:54](#), [05:99](#), [05:100](#), [05:101](#), [05:102](#),  
   [05:103](#), [37:119](#), [37:158](#), [06:204](#),  
   [06:291](#), [06:330](#), [06:358](#), [50:353](#),  
   [50:367](#), [50:742](#), [50:753](#), [50:755](#),  
   [50:757](#), [50:768](#), [50:954](#), [50:955](#),  
   [50:956](#), [50:957](#), [50:967](#), [50:968](#),  
   [50:970](#), [50:971](#), [50:972](#), [50:974](#),  
   [50:976](#), [50:1062](#), [50:1063](#), [50:1065](#),  
   [50:1066](#), [50:1067](#), [50:1069](#), [50:1071](#),  
   [50:1089](#), [50:1090](#), [50:1091](#), [50:1092](#),  
   [50:1174](#), [50:1191](#), [50:1192](#), [50:1264](#),  
   [50:1281](#), [50:1320](#), [50:1395](#), [50:1414](#),  
   [50:1453](#), [50:1508](#), [50:1542](#), [50:1658](#),  
   [50:1660](#), [50:1743](#), [50:1746](#), [50:1759](#),  
   [50:1761](#), [52:488](#), [53:99](#), [53:175](#),  
   [53:371](#), [53:477](#), [53:478](#), [53:479](#),  
   [53:480](#), [54:582](#), [54:1998](#), [54:2035](#),  
   [57:300](#), [57:301](#), [57:302](#), [57:304](#), [1220](#)  
 \mho . . . . . [29:724](#)  
 \mid . . . . . [30:409](#)  
 \min . . . . . [38:23](#)  
 minipage (env.) . . . . . [819](#)  
 \minipage . . . . . [40:424](#)  
 \mit . . . . . [29:776](#)  
 \mkern . . . . . [30:342](#),  
   [30:345](#), [30:347](#), [30:469](#), [30:478](#),  
   [30:520](#), [30:521](#), [30:522](#), [30:552](#),  
   [30:553](#), [30:554](#), [30:555](#), [30:556](#),  
   [30:557](#), [30:558](#), [30:559](#), [38:36](#), [38:37](#),  
   [38:40](#), [38:73](#), [38:74](#), [44:215](#), [44:238](#)  
 mlist commands:  
   mlist\_to\_hlist . . . . . [04:1008](#)  
 mode commands:  
   \mode\_if\_horizontal:TF . [16:95](#), [16:100](#)  
   \mode\_if\_inner:TF . . . . . [16:96](#)  
   \mode\_if\_math:TF . . . . . [28:344](#)  
   \mode\_if\_vertical:TF . [16:114](#), [16:126](#)  
 \models . . . . . [30:491](#), [1239](#)  
 module commands:  
   module\_error . . . . . [04:344](#)  
   module\_info . . . . . [04:344](#)  
   module\_warning . . . . . [04:344](#)  
   \module\_error . . . . . [45](#)  
   \module\_info . . . . . [45](#)  
   \module\_warning . . . . . [45](#)  
 modules . . . . . [04:297](#)

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\month 03:17, 01:169,
 50:1298, 50:1431, 50:1520, 02:385
\moveright 54:629, 54:688
\mp 30:396
\mscount 41:379
msg commands:
 \msg_... 1259
 \msg_error:n 10:49, 11:267,
 16:122, 16:132, 36:14, 08:436, 08:453
 \msg_error:nmn 08:2147, 10:30, 10:92, 10:113,
 10:130, 11:62, 11:69, 11:185, 11:233,
 11:247, 11:291, 11:313, 11:413,
 11:426, 11:443, 11:499, 11:537,
 11:616, 11:625, 11:731, 11:883,
 48:11, 48:214, 48:244, 51:244, 52:75,
 07:345, 07:746, 07:2862, 07:2868,
 07:2879, 07:2898, 07:2901, 07:2909,
 08:76, 08:94, 08:179, 08:208, 08:211,
 08:461, 08:525, 08:542, 08:596, 08:616
 \msg_error:nnnn 09:43, 09:101, 09:397,
 10:78, 10:109, 10:127, 11:56, 11:75,
 11:199, 11:555, 11:583, 11:595,
 11:684, 11:696, 11:796, 11:900,
 11:906, 16:20, 16:33, 16:53, 16:65,
 16:108, 36:35, 51:38, 51:74, 51:218,
 07:459, 07:467, 07:491, 07:495,
 07:531, 07:562, 07:577, 07:659,
 07:666, 07:675, 07:703, 07:715,
 07:734, 07:768, 07:1115, 07:1820,
 07:2429, 07:2433, 07:2810, 07:2824,
 07:2929, 07:2943, 08:112, 08:507, 08:640
 \msg_error:nnnnn 11:48, 11:410, 07:690, 07:1964,
 07:1971, 07:2268, 08:440, 08:471, 08:481
 \msg_error:nnnnnn 08:1321, 08:1333, 08:1358, 08:1760
\msg_expandable_error:nn 08:341, 08:342
\msg_expandable_error:nnn 08:2542, 36:77, 36:113,
 05:185, 07:416, 07:2975, 08:370, 08:1261
\msg_expandable_error:nnnn 52:464, 07:2160, 07:2187
\msg_info:nnnn 11:109, 11:193, 11:396,
 11:641, 11:664, 11:680, 07:69,
 07:80, 07:230, 07:231, 07:247, 07:251
\msg_line_context: 08:2624, 08:2629, 08:2633, 08:2680,
 11:1006, 11:1012, 11:1019, 11:1026,
 11:1033, 11:1040, 11:1048, 11:1135,
 11:1137, 11:1141, 11:1144, 48:20,
 48:92, 48:165, 48:230, 07:2767,
 07:2772, 07:2777, 07:2782, 08:563
\msg_module_name:n 51:93, 51:223, 51:229, 51:233
\g_msg_module_name_prop 08:2566
\g_msg_module_type_prop 08:2564,
 08:2565, 10:169, 11:1155, 48:279, 07:54
\msg_new:nnn 08:2622, 08:2627, 08:2631, 08:2676,
 08:2682, 08:2687, 08:2692, 08:2697,
 08:2701, 08:2708, 08:2873, 11:1134,
 11:1136, 11:1138, 11:1143, 11:1145,
 11:1150, 05:187, 51:221, 52:470,
 07:1116, 07:2721, 07:2764, 07:2769,
 07:2774, 07:2779, 07:2784, 07:2796
\msg_new:nnnn 08:2567, 08:2577, 08:2584,
 08:2591, 08:2600, 08:2614, 08:2636,
 08:2652, 08:2665, 09:575, 09:585,
 10:146, 10:151, 10:155, 10:159,
 10:165, 11:962, 11:970, 11:976,
 11:985, 11:991, 11:998, 11:1005,
 11:1011, 11:1018, 11:1025, 11:1032,
 11:1039, 11:1047, 11:1054, 11:1063,
 11:1069, 11:1075, 11:1082, 11:1088,
 11:1094, 11:1100, 11:1106, 11:1122,
 11:1128, 16:144, 16:155, 36:245,
 36:251, 36:257, 48:280, 48:288,
 48:296, 51:89, 51:226, 51:249,
 07:2557, 07:2564, 07:2571, 07:2578,
 07:2585, 07:2592, 07:2599, 07:2606,
 07:2613, 07:2620, 07:2628, 07:2635,
 07:2642, 07:2649, 07:2657, 07:2665,
 07:2674, 07:2681, 07:2688, 07:2695,
 07:2702, 07:2708, 07:2714, 07:2723,
 07:2729, 07:2736, 07:2743, 07:2754
\msg_redirect_module:nnn 07:53
\msg_show:nnnnn 11:953
\msg_show:nnnnnn 11:942
\msg_show_item_unbraced:nn 11:946, 11:958
\msg_warning:nnn 07:77, 08:295, 08:313
\msg_warning:nnnn 08:2872, 51:217,
 07:727, 08:978, 08:991, 08:1013, 08:1027
\msg_warning:nnnnn 08:768
\msg_warning:nnnnnn 08:1768
msg internal commands:
 \c_msg_coding_error_text_t1 48:283, 48:291, 48:299
 \c_msg_return_text_t1 48:286, 48:302
\mskip 38:36, 38:38,
 38:219, 38:238, 38:241, 38:243, 38:244
\mu 30:286
\mubyte 57:352

```

**File Key:** 01=ltidirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\multicolumn 41:233, 1167
\multiput 42:78, 42:810, 42:827
\multispan 41:233, 41:379, 1228
\muskip 04:34, 30:552, 30:553, 02:29, 02:55, 02:93
muskip commands:
 \muskip_eval:n 11:366
 \muskip_new:N 11:35
 \l_tmpa_muskip 346
\muskipdef 04:228, 02:55, 02:93
\muskipzero 04:228

N
\n 04:331, 04:333, 04:340, 04:342, 04:469, 04:688, 04:713, 04:737, 04:778, 04:800, 04:819, 04:827, 04:828, 04:853, 04:870, 04:909, 04:916, 04:917, 04:924, 04:936, 57:121, 57:126
\nabla 30:326
\nAME 81
\narrower 02:526
\natural 30:336
\ncallback 04:783
\ndefault 04:788, 04:792
\neg 30:420, 675
\nearrow 30:412
\NeedsFormat 1198
\NeedsTeXFormat 26:12, 33:817, 50:747, 50:1805, 1202
\neg 30:332, 30:333
\negmedspace 38:214, 1254
\negthickspace 38:214, 1254
\negthinspace 18:544, 38:214, 1251
\neq 30:419, 675
new commands:
 new_attribute 04:410
 new_bytocode 04:444
 new_chunkname 04:457
 new_luafunction 04:473
 new_whatsit 04:432
 \new_attribute 43
 \new_bytocode 43
 \new_chunkname 43
 \new_luafunction 43
 \new_whatsit 43
 \newattribute 42
 \newattribute 04:74, 04:238
\newbox ... 12:13, 37:583, 38:66, 39:27, 40:154, 41:16, 41:17, 41:18, 41:343, 42:6, 42:669, 42:674, 54:86, 54:122, 54:123, 54:124, 02:47, 02:314, 02:523
\newcatcodetable 42
\newcatcodetable 04:84, 04:93, 04:94, 04:120, 04:121, 04:242
\newcommand 81
\newcommand 21:4, 25:530, 25:535, 25:540, 29:36, 29:72, 30:59, 30:60, 30:61, 30:62, 30:64, 30:65, 30:67, 30:68, 30:100, 30:101, 30:102, 30:103, 30:104, 30:105, 30:127, 30:128, 30:129, 36:240, 37:396, 37:397, 37:398, 37:399, 06:77, 42:681, 52:573, 52:574, 52:575, 52:576, 52:578, 54:2329, 54:2332, 54:2335, 54:2336, 54:2339, 54:2340, 57:561, 100
\NewCommandCopy 06:473, 06:475, 07:1092, 07:1144, 07:1162, 07:1175, 07:1190, 07:1308, 98
\newcount 12:7, 12:8, 18:125, 20:7, 22:41, 22:61, 26:25, 28:27, 28:142, 28:429, 38:55, 38:357, 38:358, 39:23, 39:24, 39:25, 39:26, 39:56, 39:250, 39:265, 40:473, 41:11, 41:12, 41:13, 41:14, 41:15, 41:335, 41:336, 41:337, 42:663, 42:664, 42:665, 42:666, 42:675, 44:36, 44:140, 44:141, 45:3, 45:267, 45:268, 45:269, 45:270, 50:1594, 54:105, 54:107, 54:109, 54:111, 54:113, 54:121, 54:2024, 54:2327, 54:2330, 54:2333, 54:2337, 57:3, 57:4, 57:5, 57:91, 02:47, 02:356, 02:427, 1241
\newcounter 512
\newcounter 22:10
\newdimen 12:10, 12:11, 12:12, 18:124, 26:398, 26:399, 38:53, 39:9, 39:10, 39:11, 39:12, 39:13, 39:14, 39:15, 39:16, 39:17, 39:18, 39:19, 39:20, 39:21, 39:22, 40:210, 40:211, 41:3, 41:5, 41:6, 41:7, 41:8, 41:166, 41:338, 41:339, 41:340, 41:341, 42:3, 42:4, 42:5, 42:7, 42:431, 42:432, 42:433, 42:434, 42:435, 42:436, 42:667, 42:668, 42:670, 42:671, 42:672, 42:673, 45:451, 54:71, 54:72, 54:73, 54:75, 54:76, 54:77, 54:78, 54:79, 54:80, 54:81, 54:82, 54:83, 54:84, 54:85, 54:91, 54:93, 54:94, 54:95, 54:96, 54:108, 54:110, 54:112, 54:114, 54:115, 54:116, 54:117, 54:118, 54:119, 54:120, 54:2025, 54:2026, 02:47, 02:309, 02:311, 02:312, 02:426
\NewDocumentCommand . 08:2713, 08:2715, 08:2717, 08:2721, 08:2723, 08:2725, 08:2736, 08:2738, 08:2748, 08:2750, 08:2759, 08:2761, 08:2768, 08:2770, 08:2772, 08:2774, 08:2776, 08:2800,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

08:2802, 08:2805, 17:24, 17:46,  
 17:55, 17:56, 17:57, 20:750, 22:127,  
 22:135, 35:66, 35:68, 35:155, 36:80,  
 51:184, 51:186, 51:197, 51:255,  
 57:627, 57:634, 57:641, 07:2804, 193  
\NewDocumentEnvironment . . . . .  
. . . . . 07:2588, 07:2595, 07:2840, 217  
\newenvironment . . . . . 82  
\newenvironment . . . . .  
06:146, 50:1296, 50:1429, 50:1518, 217  
\NewEnvironmentCopy . 06:514, 06:516, 1264  
\NewExpandableDocumentCommand . . . . .  
. . . . . 36:118, 48:366, 48:368, 48:370,  
48:372, 48:375, 48:378, 57:597, 07:2923  
\newfam . . . . . 04:38, 24:16, 02:47, 1225  
\newfont . . . . . 29:618  
\newgroup . . . . . 28:47  
\newhelp . . . . . 02:306  
\NewHook . . . . . 08:2713,  
08:2888, 20:73, 20:74, 20:75, 20:348,  
20:351, 20:400, 26:150, 29:420,  
29:421, 29:422, 29:423, 29:424,  
29:425, 29:426, 29:427, 29:428,  
37:34, 37:35, 37:36, 37:37, 37:38,  
37:99, 37:100, 37:101, 37:102,  
37:103, 50:1040, 50:1041, 52:177, 227  
\NewHookPair . . . . . 305  
\NewHookWithArguments 08:2719, 35:101, 195  
\newif . . . . . 03:73, 12:9, 20:5,  
20:6, 24:204, 25:401, 25:413, 28:15,  
29:528, 32:82, 33:868, 33:1221,  
33:1222, 35:3, 38:75, 38:76, 38:203,  
38:359, 39:28, 39:29, 39:30, 39:31,  
39:32, 39:33, 39:140, 40:423, 40:529,  
41:19, 41:251, 42:157, 42:427,  
42:428, 42:429, 42:430, 42:459,  
42:460, 06:168, 44:38, 44:124, 50:2,  
54:97, 54:98, 54:99, 54:100, 54:101,  
54:102, 54:103, 54:104, 06:872, 1193  
\newinsert . . . . . 40:474, 45:390,  
54:27, 54:1856, 02:193, 02:242, 1244  
\newlabel . . . . . 35:84, 35:106, 35:166, 1227  
\newlanguage . . . . . 57:284, 02:47  
\newlength . . . . . 521  
\newlength . . . . . 23:3  
\newline . . . . . 18:93, 18:100, 18:106  
\newlinechar . . . . . 06:20, 01:56, 02:381  
\newluabytecode . . . . . 43  
\newluabytecode . . . . . 04:197, 04:252  
\newluachunkname . . . . . 43  
\newluachunkname . . . . . 04:205, 04:254  
\newluacmd . . . . . 42  
\newluacmd . . . . . 04:181, 43  
\newluafunction . . . . . 42  
\newluafunction . . . . .  
. . . . . 04:4, 04:173, 04:236, 04:248, 42  
\NewMarkClass . . . . .  
48:363, 48:482, 57:25, 57:26, 57:27, 943  
\newmarks . . . . . 48:22, 57:6, 948  
\newmathalphabet . . . . . 27:13, 27:109  
\NewMirroredHookPair 08:2713, 08:2890, 195  
\NewMirroredHookPairWithArguments . . . . .  
. . . . . 08:2719, 195  
\NewModuleRelease . . . . .  
09:4, 10:4, 11:7, 03:152, 16:4, 17:3,  
33:2, 36:4, 48:4, 55:242, 07:9, 08:4, 1257  
\newmuskip . . . . . 02:47  
\newpage . . . . . 54:135, 54:141, 54:152, 328  
\NewProperty . . . . . 36:38, 36:267, 749  
\newprotectedluacmd . . . . . 43  
\newprotectedluacmd 04:181, 21:1010, 53:28  
\newread . . . . . 02:47, 02:307, 1196  
\NewReversedHook . . . . . 08:2713,  
08:2889, 20:349, 20:350, 20:401,  
20:402, 50:1042, 50:1043, 52:178, 195  
\NewReversedHookWithArguments . . . . .  
. . . . . 08:2719, 195  
\newrobustcmd . . . . . 96  
\newsavebox . . . . . 818  
\newsavebox . . . . . 40:154  
\newskip . . . . . 12:14, 12:15, 12:17,  
18:450, 18:451, 18:452, 18:517,  
18:530, 23:3, 37:493, 38:360, 38:460,  
39:2, 39:3, 39:4, 39:5, 39:6, 39:7,  
39:8, 54:2341, 54:2342, 54:2343,  
54:2347, 54:2348, 54:2351, 54:2352,  
54:2353, 54:2357, 54:2358, 54:2359,  
02:47, 02:310, 02:313, 02:424, 02:425  
\NewSocket . . . . . 10:170,  
10:182, 35:117, 35:118, 55:39, 55:49,  
55:63, 55:64, 55:65, 55:66, 55:67,  
55:68, 55:69, 55:70, 55:71, 55:72,  
55:73, 55:74, 55:75, 55:76, 55:77,  
55:78, 55:79, 55:80, 55:81, 55:82,  
55:83, 55:84, 55:85, 55:86, 55:87,  
55:88, 55:89, 55:90, 55:91, 55:92,  
55:93, 55:94, 55:95, 55:96, 55:97, 334  
\NewSocketPlug . . . . . 10:161, 10:170,  
10:186, 35:119, 55:40, 55:51, 330  
\NewTemplateType . . . . .  
. . . . . 11:1066, 11:1125, 11:1156, 343  
\newtheorem . . . . . 43:1, 1228  
\newtie . 21:851, 33:188, 33:189, 33:208,  
33:705, 33:1012, 33:1013, 33:1320, 701  
\newtoks . . . . .  
03:36, 12:16, 16:77, 24:351, 24:352,  
25:499, 26:247, 02:47, 02:306, 1244  
\newwhatsit . . . . . 43

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

\newwhatsit . . . . . 04:189, 04:250  
\newwrite . . . . . 20:3, 20:4,  
    44:154, 46:4, 46:21, 02:47, 02:308, 1218  
\newXeTeXintercharclass . . . . . 57:35  
\next . . . . . 1221  
\NextLinkTarget . . . . . 17:20  
nfss internal commands:  
    \\_\\_nfss\\_init\\_mv\\_freeze:N . . . . .  
        . . . . . 28:318, 28:343, 28:344  
\NG . . . . . 21:531, 21:1145, 57:650, 1213  
\ng . . . . . 21:552, 21:1146, 57:650, 1213  
\ni . . . . . 30:437, 30:438  
\noalign . . . . . 30:346, 30:538, 30:541, 30:544,  
    30:545, 30:549, 30:550, 38:171,  
    38:172, 38:188, 38:191, 38:205,  
    38:404, 38:414, 41:224, 41:230,  
    41:359, 41:378, 42:148, 42:154, 1193  
\nobreak . . . . . 18:60, 18:72, 18:116, 18:142,  
    18:148, 18:161, 18:174, 18:200,  
    18:398, 18:406, 18:432, 18:440,  
    18:461, 18:468, 18:528, 20:203,  
    20:215, 21:409, 21:435, 21:437,  
    21:568, 21:1165, 37:418, 37:425,  
    40:599, 40:609, 44:90, 44:212,  
    44:213, 44:217, 44:235, 44:236,  
    44:240, 45:531, 48:241, 49:33,  
    49:44, 49:58, 49:66, 54:338, 54:1153,  
    54:1319, 06:886, 06:907, 57:208,  
    57:210, 57:214, 57:215, 57:216,  
    57:220, 02:503, 02:506, 02:508, 1224  
\nobreakdashes . . . . . 18:453  
\nobreakspace . . . . . 18:467, 438  
\nobreakspace\_ . . . . . 1233  
\NoCaseChange . . . . . 57:571, 1262  
\nocide . . . . . 940  
\nocite . . . . . 937  
\nocite . . . . . 47:59, 1220  
\nocorr . . . . . 32:43, 32:58, 32:62, 32:65, 1208  
\nocorrlist . . . . . 32:89, 32:121  
\noexpand . . . . . 1195  
\nofiles . . . . . 442  
\nofiles . . . . . 20:199, 1222  
\noident . . . . . 24:694, 24:720, 44:139, 406  
\nointerlineskip . . . . . 30:346,  
    30:538, 30:541, 30:545, 30:549,  
    38:297, 38:323, 42:562, 42:565,  
    42:575, 42:577, 54:1846, 54:1854, 02:498  
\nolimits . . . . . 30:355, 30:362,  
    38:3, 38:4, 38:5, 38:9, 38:10, 38:11,  
    38:12, 38:13, 38:14, 38:15, 38:16,  
    38:17, 38:18, 38:19, 38:20, 38:21,  
    38:26, 38:27, 38:28, 38:29, 38:31, 38:34  
\nolinebreak . . . . . 421  
\nolinebreak . . . . . 18:9, 18:28  
\nonfrenchspacing . . . . . 02:709, 20:48,  
    20:118, 20:176, 06:887, 06:908, 02:431  
\nonscript . . . . . 38:36, 38:38  
\nonstopmode . . . . . 955  
\nonumber . . . . . 38:387, 38:426, 38:427  
\nopagebreak . . . . . 421  
\nopagebreak . . . . . 18:7, 18:26  
\noprotrusion . . . . . 44:222, 44:245  
\normalbaselines . . . . . 38:167,  
    38:169, 02:396, 02:409, 02:410, 02:435  
\normalbaselineskip . . . . . 26:188, 40:396, 40:415, 02:424, 02:436  
\normalcolor . . . . .  
    . . . . . 38:352, 38:456, 40:110, 40:465,  
    44:218, 44:241, 45:97, 45:166,  
    54:218, 54:496, 54:633, 54:643,  
    54:692, 54:702, 54:2264, 54:2297, 1217  
\normalfont . . . . . 24:671, 24:732, 29:653,  
    29:673, 29:675, 29:684, 29:689,  
    29:691, 29:699, 29:706, 29:708,  
    29:714, 32:18, 37:571, 38:352,  
    38:456, 44:218, 44:241, 45:401, 1226  
normalfont . . . . . 29:420  
\normalize . . . . . 1206  
\normallineskip . . . . .  
    . . . . . 40:394, 40:414, 02:424, 02:435  
\normallineskiplimit . . . . . 38:206,  
    40:379, 40:395, 40:401, 02:424, 02:436  
\normalmarginpar . . . . . 45:387  
\normalsfcodes . . . . .  
    . . . . . 20:44, 20:46, 20:48, 20:114,  
    20:116, 20:118, 20:172, 20:174,  
    20:176, 20:198, 54:622, 54:681, 1235  
\normalshape . . . . . 25:682, 25:723, 570  
\normalsize . . . . . 20:42, 20:112,  
    20:170, 32:142, 45:23, 45:176,  
    45:372, 50:5, 54:621, 54:680, 198  
\not . . . . . 30:345, 30:419, 30:420, 30:442  
\notexpanded . . . . . 318  
\notin . . . . . 30:468  
\noxpand . . . . . 1198  
\nu . . . . . 30:287  
\null . . . . . 21:327,  
    21:363, 21:486, 21:489, 21:834,  
    21:837, 21:1253, 35:21, 35:38, 35:55,  
    37:536, 37:557, 37:720, 37:730,  
    38:112, 38:121, 38:169, 38:198,  
    44:212, 44:235, 53:385, 02:449, 1223  
\nulldelimiterspace . . . . . 30:649, 02:398  
\nullfont . . . . . 37:242  
\number . . . . . 03:52, 03:61,  
    04:105, 22:190, 24:621, 24:624,  
    26:439, 28:64, 28:93, 28:113, 28:128,  
    28:164, 28:195, 28:225, 28:257,

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefsns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfnctcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltbypnem.dtx, 57=ltfinal.dtx

29:645, 06:2, 06:114, 50:1205,  
 50:1298, 50:1431, 50:1520, 53:351, 01:70  
 $\backslash$ numberline 44:72, 44:82, 44:248, 45:17, 1197  
 $\backslash$ numexpr ..... 04:82, 04:105,  
 04:157, 21:1034, 28:150, 54:36,  
 06:747, 02:189, 02:205, 02:215, 02:246  
 $\backslash$ nunknown ..... 04:805  
 $\backslash$ narrow ..... 30:414  
 $\backslash$ nxt ..... 1221

**O**

$\backslash$ o ..... 21:244,  
 21:400, 21:533, 21:770, 21:1130, 57:649  
 $\backslash$ o ..... 21:253,  
 21:405, 21:554, 21:781, 21:1136, 57:649  
 $\backslash$ oalign ..... 02:535  
 $\backslash$ obeycr ..... 18:556  
 $\backslash$ obeyedline ..... 07:1979, 07:1988,  
 07:1995, 02:459, 02:463, 02:486, 28  
 $\backslash$ obeyedspace ... 02:464, 02:467, 02:487, 29  
 $\backslash$ obeylines 37:542, 37:563, 37:711, 37:712,  
 54:587, 06:888, 06:909, 02:452, 28  
 $\backslash$ obeyspaces .....  
 ... 54:587, 06:889, 06:910, 02:452, 29  
 $\backslash$ oddsidemargin 54:72, 54:74, 54:615, 54:674  
 $\backslash$ odot ..... 30:391  
 $\backslash$ OE ..... 21:243,  
 21:399, 21:532, 21:769, 21:1147,  
 57:607, 57:612, 57:617, 57:649, 1189  
 $\backslash$ oe ..... 21:252,  
 21:404, 21:553, 21:782, 21:1148,  
 57:607, 57:612, 57:617, 57:649, 1264  
 $\backslash$ of ..... 38:67, 38:356  
 $\backslash$ offinterlineskip ..... 02:498  
 $\backslash$ oint ..... 30:362  
 $\backslash$ ointop ..... 30:361, 30:362  
 $\backslash$ oldstylenums ..... 33:4,  
 33:376, 33:377, 33:378, 33:379,  
 33:380, 33:381, 33:382, 33:383,  
 33:384, 33:385, 33:610, 33:1133, 476  
 $\backslash$ Omega ..... 30:314  
 $\backslash$ omega ..... 30:297  
 $\backslash$ ominus ..... 30:394  
 $\backslash$ omit ..... 38:191,  
 38:192, 41:369, 41:372, 41:379, 41:383  
 $\backslash$ 0mitIndent ..... 16:91, 16:170, 412  
 $\backslash$ onecolumn ..... 54:143  
 $\backslash$ OnlyDescription ..... 26:5, 31:3  
 $\backslash$ ooalign ..... 21:327, 21:357,  
 21:394, 21:480, 21:486, 21:488,  
 21:499, 21:515, 21:731, 21:764,  
 21:834, 21:837, 21:894, 21:1253,  
 29:650, 30:469, 30:472, 02:535, 1236  
 $\backslash$ openin ..... 1250

**P**

$\backslash$ P ..... 21:310, 1218  
 $\backslash$ package/.../after ..... 1037  
 $\backslash$ package/.../before ..... 1037  
 $\backslash$ package/after ..... 1037  
 $\backslash$ package/before ..... 1037  
 $\backslash$ PackageError ... 03:77, 03:135, 03:147,  
 14:84, 21:1547, 33:823, 33:874, 33:918  
 $\backslash$ PackageInfo ... 14:84, 33:827, 33:843,  
 33:848, 33:864, 33:865, 33:925, 33:1210  
 $\backslash$ PackageNote ..... 14:136, 1258  
 $\backslash$ PackageNoteNoLine ..... 14:136  
 $\backslash$ PackageWarning .....  
 ... 14:84, 33:825, 33:875, 33:1139, 53:476  
 $\backslash$ PackageWarningNoLine .....  
 ... 14:84, 21:986, 54:1997  
 $\backslash$ page ..... 36:234, 750  
 $\backslash$ pagebreak ..... 421  
 $\backslash$ pagebreak ..... 18:6, 18:7, 18:23, 18:25  
 $\backslash$ pagegoal ..... 54:1884, 54:1891  
 $\backslash$ pagenum ..... 36:235, 750  
 $\backslash$ pagenumbering ..... 737  
 $\backslash$ pagenumbering ..... 34:5  
 $\backslash$ pageref ..... 35:10, 746  
 $\backslash$ pageshrink ..... 54:542, 54:546, 54:562  
 $\backslash$ pagestyle ..... 49:2  
 $\backslash$ pagetarget ..... 751  
 $\backslash$ pagetotal ..... 54:130  
 $\backslash$ paperheight ..... 54:93

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\paperwidth ..... 54:93  
 \par ... 01:104, 15:3, 15:4, 15:5, 16:140,  
   16:176, 18:239, 18:290, 04:156,  
   24:697, 37:184, 37:240, 37:418,  
   37:425, 37:534, 37:555, 39:63, 06:9,  
   39:110, 39:127, 39:130, 39:137,  
   06:21, 39:185, 39:188, 40:385,  
   40:406, 40:461, 40:491, 40:508,  
   41:195, 41:385, 44:41, 44:90, 44:220,  
   44:242, 45:15, 45:24, 45:249, 45:344,  
   45:478, 45:496, 49:91, 49:92, 53:401,  
   02:11, 54:168, 54:195, 54:259,  
   54:1890, 02:445, 02:461, 02:463,  
   02:480, 02:502, 02:511, 02:512,  
   02:513, 02:515, 02:517, 02:519, 402  
 para commands:  
   \para\_end: ..... 16:93,  
     16:93, 16:140, 16:141, 16:142, 413  
   \g\_para\_indent\_box .. 16:16, 16:49,  
     16:84, 16:86, 16:89, 16:91, 16:117, 409  
   \para OMIT indent: .. 16:88, 16:92, 406  
   \para\_raw\_end: .....  
     ..... 16:113, 16:135, 16:138, 406  
   \para\_raw\_indent: .....  
     ..... 16:113, 16:113, 16:136, 406  
   \para\_raw\_noindent: .....  
     ..... 16:113, 16:125, 16:137, 406  
 para internal commands:  
   \\_\\_para\_handle\_indent: .....  
     ... 16:34, 16:66, 16:85, 16:85, 16:119  
   \g\\_para\_standard\_everypar\_tl ...  
     16:12, 16:76, 16:78, 16:118, 16:129, 411  
   \\_\\_para\_tmp:w 16:37, 16:41, 16:68, 16:72  
 para/after ..... 16:6, 404  
 para/before ..... 16:6, 404  
 para/begin ..... 16:6, 404  
 para/end ..... 16:6, 404  
 \paracntvalue ..... 407  
 \paragraphmark ..... 44:143  
 \parallel ..... 30:408  
 \parbox ..... 818  
 \parbox ..... 40:298, 402  
 \parfillskip ... 24:670, 24:686, 24:731,  
   37:454, 37:464, 37:473, 37:481,  
   37:530, 37:552, 39:76, 40:393,  
   40:414, 44:207, 44:230, 02:423, 415  
 \parindent 37:454, 37:459, 37:464, 37:473,  
   37:477, 37:481, 37:530, 37:552,  
   39:50, 40:388, 40:409, 44:208,  
   44:231, 02:404, 02:527, 02:528, 401  
 \parsep ..... 39:1, 39:49, 39:90, 803  
 \parseunicodedataI ..... 04:123, 04:162  
 \parseunicodedataII ..... 04:124, 04:126  
 \parseunicodedataIII ..... 04:128, 04:134  
   \parseunicodedataIV ..... 04:130, 04:142  
   \parseunicodedataV ..... 04:146, 04:149  
   \parshape ..... 39:54, 409  
   \parskip ..... 37:429, 37:528, 37:530,  
     37:550, 37:552, 38:528, 39:49,  
     39:73, 39:88, 39:90, 39:117, 39:177,  
     39:196, 39:247, 40:388, 40:409,  
     41:79, 54:1163, 54:1331, 02:411, 410  
   \partial ..... 30:322  
   \partopsep ..... 38:526, 39:1, 39:61, 803  
   \PassOptionsToClass ..... 976  
   \PassOptionsToClass ..... 50:441  
   \PassOptionsToPackage ..... 976  
   \PassOptionsToPackage ..... 50:441  
   \patterns ..... 21:205, 1217  
   \pausing ..... 02:338  
   \pdffilesize ..... 05:71, 05:117  
   \pdfgentounicode ..... 57:315, 57:316,  
     57:320, 57:335, 57:340, 57:341, 57:342  
   \pdfhorigin ..... 53:314  
   \pdfsavepos ..... 751  
   \pdftexrevision ..... 57:321  
   \pdftexversion ..... 57:319, 57:320, 57:321  
   \pdfvariable ..... 53:313, 53:318  
   \pdfvorigin ..... 53:319  
 peek commands:  
   \peek\_meaning:NTF ..... 07:2518, 182  
   \peek\_meaning\_remove:NTF .....  
     ..... 07:1861, 07:1962, 07:2506, 182  
   \peek\_N\_type:TF .....  
     ..... 07:1867, 07:1891, 07:1923  
   \peek\_remove\_spaces:n ..... 07:1859  
 \penalty ..... 18:35,  
   18:38, 18:47, 18:296, 18:306, 18:329,  
   18:339, 18:363, 18:367, 32:118,  
   37:536, 37:539, 37:557, 37:560,  
   38:37, 38:207, 38:404, 38:414,  
   39:214, 41:56, 45:195, 45:199,  
   45:201, 45:217, 45:221, 45:223,  
   47:37, 54:138, 54:178, 54:197,  
   54:200, 54:1161, 54:1327, 02:507,  
   02:508, 02:509, 02:510, 02:511,  
   02:512, 02:516, 02:518, 02:520, 1204  
   \perp ..... 30:454  
   \phantom ..... 38:75  
   \Phi ..... 30:312  
   \phi ..... 30:294  
   \Pi ..... 30:309  
   \pi ..... 30:289  
 picture (env.) ..... 42:21  
 \picture ..... 42:21  
 Plugs:  
   default ..... 55:40  
   kernel (refstepcounter/target) 35:119

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

kernel (tagsupport/recordtarget)
 55:50
\pm 30:397
\pmatrix 38:173, 38:181
\pmod 38:39
\PopDefaultHookLabel 08:2772, 201
\poptabs 14:256, 41:142, 41:161
\poptracing 26:148, 26:340
\postdisplaypenalty 18:13, 38:468, 38:480, 38:506, 02:330
\pounds 21:318, 1214
\Pr 38:32
pre commands:
 pre_shipout_filter 1063
\prec 30:428
\preceq 30:431
\predisplaypenalty 18:12,
 38:467, 38:479, 38:505, 02:329, 1264
\predisplaysize 02:401
\pretolerance 24:672, 24:688, 24:733, 02:316
\prevdepth 18:302, 18:303, 18:335,
 18:336, 18:396, 18:401, 18:430,
 18:435, 38:205, 45:196, 45:198,
 45:218, 45:220, 54:169, 54:171,
 54:174, 02:498, 02:502, 02:503, 835
\PreviousTotalPages 53:407, 1064
prg commands:
 \prg_break:n 07:2965,
 07:2967, 07:2969, 07:2971, 07:2972
 \prg_break_point: 07:2973
 \prg_do_nothing: 08:2298, 09:193, 09:268,
 09:387, 09:606, 48:236, 52:132,
 52:134, 53:166, 53:167, 53:172,
 53:181, 53:185, 53:321, 55:22,
 55:32, 07:407, 07:959, 07:1803, 141
 \prg_generate_conditional_-
 variant:Nnn 11:83, 36:165, 36:180, 36:195
 \prg_new_conditional:Npnn 08:2413, 08:2428, 08:2445,
 08:2462, 08:2468, 08:2474, 08:2480,
 08:2489, 08:2497, 08:2511, 08:2529,
 09:311, 11:77, 11:84, 11:90, 36:154,
 36:169, 36:184, 48:265, 48:272,
 55:10, 08:273, 08:881, 08:1194, 08:1404
 \prg_new_protected_conditional:Npnn
 09:107,
 09:450, 08:732, 08:790, 08:820, 08:852
 \prg_replicate:nn 08:2281, 08:2289, 08:2359, 08:2375,
 52:552, 07:877, 07:920, 07:970,
 07:1025, 07:2275, 07:2282, 08:1168

```

\prg\_return\_false: ..... 08:2423, 08:2439, 08:2457,
 08:2466, 08:2472, 08:2478, 08:2486,
 08:2495, 08:2504, 08:2507, 08:2521,
 08:2536, 09:122, 09:324, 09:465,
 11:81, 11:88, 11:94, 36:162, 36:177,
 36:192, 48:270, 48:277, 55:11,
 08:279, 08:759, 08:813, 08:825,
 08:845, 08:857, 08:872, 08:889,
 08:893, 08:896, 08:1199, 08:1409, 246

\prg\_return\_true: ..... 08:2421,
 08:2437, 08:2456, 08:2459, 08:2465,
 08:2471, 08:2477, 08:2484, 08:2494,
 08:2505, 08:2519, 08:2534, 09:121,
 09:324, 09:464, 11:80, 11:87, 11:93,
 36:159, 36:174, 36:189, 48:269,
 48:276, 08:278, 08:750, 08:804,
 08:843, 08:870, 08:892, 08:1197, 08:1407

\prime ..... 30:260, 30:324, 38:256

\ProcessedArgument ..... 07:388,
 07:392, 07:399, 07:2208, 07:2209,
 07:2226, 07:2228, 07:2250, 07:2259,
 07:2265, 07:2273, 07:2290, 07:2302,
 07:2329, 07:2395, 07:2397, 07:3000, 178

\ProcessKeyOptions ... 51:63, 51:197, 1262
\ProcessKeyPackageOptions ..... 1261
\ProcessList ..... 07:3005
\ProcessOption\* ..... 1198
\ProcessOptions 21:1572, 26:71, 33:841,
 33:878, 50:537, 50:628, 50:1193, 1204
\ProcessOptions\* ..... 50:537
\prod ..... 30:356
prop commands:
 \prop\_clear:N ..... 11:144, 11:154, 11:170, 11:180,
 11:209, 11:210, 11:405, 07:1711, 08:1141
 \prop\_clear\_new:N ..... 11:123
 \prop\_const\_from\_keyval:Nn .....
 .. 08:935, 08:943, 08:945, 08:947
 \prop\_gclear:N .... 09:58, 11:110,
 08:968, 08:1003, 08:1425, 08:1452
 \prop\_gclear\_new:N ..... 08:2400, 08:2410, 11:99, 11:131
 \prop\_get:NnN ..... 11:44,
 11:550, 11:553, 11:580, 11:691,
 11:712, 11:793, 08:565, 08:1595, 08:1681
 \prop\_get:NnNTF .. 11:418, 11:531,
 11:622, 11:717, 11:719, 11:804,
 11:877, 51:202, 07:1717, 08:574, 08:619
 \prop\_gpop:NnNTF ..... 08:976, 08:1011
 \prop\_gput:Nnn ..... 08:2564, 08:2565, 08:2566,
 09:51, 10:169, 11:195, 11:1155,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,
56=lthyphen.dtx, 57=ltfinal.dtx

48:279, 55:55, 07:54, 08:575, 08:576,  
 08:622, 08:625, 08:1502, 08:1540  
 \prop\_gset\_eq:NN .....  
     .... 11:100, 11:113, 11:132, 08:1143  
 \prop\_if\_empty:NTF .....  
     ... 08:2433, 08:1482, 08:1517, 08:1972  
 \prop\_if\_empty\_p:N .. 08:2450, 08:1868  
 \prop\_if\_exist:NTF .....  
     .... 08:2076, 08:2432, 08:2470,  
         11:107, 11:138, 11:148, 11:164, 11:174  
 \prop\_if\_exist\_p:N ..... 08:1867  
 \prop\_if\_in:NnTF ... 11:67, 11:184,  
     11:544, 11:548, 08:827, 08:835,  
     08:839, 08:859, 08:862, 08:866, 08:891  
 \prop\_map\_break: .....  
     .... 08:2084, 08:1572, 08:1654  
 \prop\_map\_function:NN .....  
     .. 09:57, 11:946, 11:957, 08:1142, 256  
 \prop\_map\_inline:Nn .....  
     .... 08:2079, 08:2081, 11:409,  
         51:237, 08:1437, 08:1458, 08:1560,  
         08:1567, 08:1569, 08:1642, 08:1649,  
         08:1651, 08:1810, 08:1871, 08:1975  
 \prop\_new:N ... 09:17, 11:18, 11:28,  
     11:30, 11:31, 11:112, 07:49, 08:30,  
     08:31, 08:152, 08:164, 08:1292, 08:1293  
 \prop\_put:Nnn 11:261, 11:353, 11:355,  
     11:360, 11:467, 11:481, 11:496,  
     11:525, 11:591, 11:660, 07:1708,  
     08:37, 08:42, 08:1177, 08:1780, 08:1791  
 \prop\_remove:Nn ..... 11:424  
 \prop\_set\_eq:NN .. 11:124, 11:141,  
     11:151, 11:167, 11:177, 08:1498, 08:1535  
 \prop\_show:N ..... 281  
 property commands:  
     \property\_gset:nnnn .....  
         .... 36:6, 36:17, 36:46, 747  
     \property\_if\_exist:n .. 36:154, 36:165  
     \property\_if\_exist:nTF .....  
         ... 36:154, 36:166, 36:167, 36:168, 749  
     \property\_if\_exist\_p:n ... 36:154, 749  
     \property\_if\_recorded:n 36:169, 36:180  
     \property\_if\_recorded:nn .....  
         .... 36:184, 36:195  
     \property\_if\_recorded:nnTF 36:184,  
         36:196, 36:197, 36:198, 36:218, 749  
     \property\_if\_recorded:nTF .....  
         .... 36:169, 36:181,  
         36:182, 36:183, 36:208, 36:216, 749  
     \property\_if\_recorded\_p:n 36:169, 749  
     \property\_if\_recorded\_p:nn 36:184, 749  
     \property\_new:nnnn .....  
         ... 36:6, 36:6, 36:41, 36:232, 36:234,

36:235, 36:236, 36:237, 36:239,  
 36:241, 36:242, 36:243, 36:244, 747  
 \property\_record:nN . 36:48, 36:48, 748  
 \property\_record:nn .....  
     ... 36:48, 36:49, 36:50, 36:52, 36:85, 748  
 \property\_ref:nn .....  
     .... 36:89, 36:89, 36:96, 36:122, 748  
 \property\_ref:nnn .....  
     .... 36:97, 36:97, 36:117, 36:125, 748  
 \property\_ref undefined\_warn: ...  
     .... 36:199, 36:199, 748  
 \property\_ref undefined\_warn:n ..  
     .... 36:206, 36:206, 748  
 \property\_ref undefined\_warn:nn ..  
     ... 36:214, 36:214, 36:230, 36:231, 747  
 property internal commands:  
     \\_\_property\_data:nnn .....  
         .... 36:128, 36:140, 36:142, 36:147  
     \\_\_property\_gset:nnnn .....  
         .... 36:6, 36:10, 36:19, 36:22, 36:37  
     \\_\_property\_record:nn .....  
         .... 36:48, 36:51, 36:53, 36:62  
     \\_\_property\_record\_value:n ....  
         .... 36:48, 36:59, 36:63  
     \\_\_property\_record\_value\_aux:n ..  
         .... 36:48, 36:64, 36:65, 36:79  
     \\_\_property\_ref:nnn .....  
         ... 36:91, 36:97, 36:99, 36:104, 36:116  
     l\_\_property\_ref\_flag ..... 36:88  
 \propto ..... 30:405  
 \protect ..... 14:246, 14:248, 14:250,  
     14:256, 14:262, 14:269, 14:279,  
     14:282, 14:288, 20:211, 21:26, 21:32,  
     21:51, 21:55, 21:209, 21:217, 28:690,  
     28:1238, 29:638, 32:143, 35:16,  
     35:33, 35:50, 37:280, 37:283, 37:317,  
     37:327, 41:264, 06:102, 44:12, 44:72,  
     44:82, 44:164, 44:171, 44:177,  
     45:17, 06:232, 06:246, 06:255,  
     06:260, 06:263, 06:264, 47:5, 06:266,  
     06:267, 06:272, 06:273, 06:278,  
     06:281, 06:282, 06:307, 06:345,  
     06:373, 52:281, 52:304, 06:580,  
     06:600, 53:83, 53:95, 53:125, 53:132,  
     53:152, 54:600, 54:659, 57:358, 1215  
     \protected .....  
         17:6, 18:57, 18:238, 18:289, 18:474,  
         18:538, 21:308, 21:309, 04:186,  
         22:242, 25:682, 25:686, 25:689,  
         25:692, 25:695, 25:698, 25:701,  
         25:704, 28:1080, 29:573, 37:201,  
         37:247, 37:417, 38:199, 38:388,  
         38:426, 38:445, 38:446, 06:7, 41:56,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

41:201, 41:208, 42:142, 06:455,  
 53:496, 53:497, 53:498, 02:463, 1261  
 \Provide ..... 1217  
 \providecommand .....  
     ... 08:2897, 08:2900, 21:6, 21:981,  
     35:98, 35:99, 35:100, 06:178, 54:2008  
 \ProvideCommandCopy ..... 98  
 \ProvideDocumentCommand ..... 07:2804  
 \ProvideDocumentEnvironment ..... 07:2840  
 \ProvideExpandableDocumentCommand .....  
     ... 07:2923  
 \ProvideHook ..... 08:2843, 1257  
 \ProvideMirroredHookPair ..... 08:2843  
 \ProvideReversedHook ..... 08:2843  
 provides commands:  
     provides\_module ..... 04:298  
 \provides\_module ..... 45  
 \ProvidesClass ..... 976  
 \ProvidesClass ..... 50:421  
 \ProvidesExplPackage ..... 52:541  
 \ProvidesFile .. 30:672, 30:674, 30:675,  
     30:676, 33:1214, 50:430, 01:73, 1220  
 \ProvidesPackage ..... 976  
 \ProvidesPackage ..... 26:13,  
     33:815, 33:846, 50:344, 50:423,  
     50:425, 50:1806, 52:570, 53:491, 1202  
 \ProvideTextCommand .... 21:3, 21:60, 1217  
 \ProvideTextCommandDefault .. 21:57, 1226  
 \Psii ..... 30:313  
 \psii ..... 30:296  
 \PushDefaultHookLabel ..... 08:2772, 202  
 \pushtabs .....  
     14:256, 41:139, 41:138, 41:158, 41:160  
 \pushtracing ..... 26:117, 26:321  
 \put ..... 42:56, 42:58, 42:70, 42:72,  
     42:325, 42:326, 42:327, 42:328,  
     42:336, 42:338, 42:351, 42:352,  
     42:353, 42:354, 42:361, 42:364,  
     42:384, 42:385, 42:386, 42:387,  
     42:393, 42:395, 42:407, 42:408,  
     42:409, 42:410, 42:415, 42:420,  
     42:728, 42:784, 42:812, 42:829, 1065

**Q**

\qbezier ..... 862  
 \qbezier ..... 42:681, 42:813, 42:830  
 \qbeziermax . 42:680, 42:706, 42:707, 42:767  
 \qqquad ..... 18:549  
 \quad ..... 18:549,  
     38:168, 38:170, 38:190, 44:111, 1224  
 quark commands:  
     \q\_mark ..... 07:1300,  
     07:1303, 07:2109, 07:2113, 07:2125,  
     07:2455, 07:2463, 07:2466, 07:2471

\q\_nil ..... 11:79,  
     11:92, 07:601, 07:1292, 07:1303,  
     07:1470, 07:1617, 07:1638, 07:1642,  
     07:1648, 07:1657, 07:1659, 07:2029,  
     07:2053, 07:2062, 07:2094, 07:2109,  
     07:2113, 07:2125, 07:2131, 07:2157, 171  
 \quark\_if\_nil:NTF 07:612, 07:2115, 171  
 \quark\_if\_nil:nTF ..... 07:1467  
 \quark\_if\_recursion\_tail\_stop:N .  
     ... 07:803, 07:1451  
 \quark\_if\_recursion\_tail\_stop:n .  
     ... 36:144, 51:7, 07:479, 07:665,  
     07:1218, 07:2468, 07:2469, 08:422  
 \quark\_if\_recursion\_tail\_stop\_-  
     do:Nn ..... 07:620  
 \quark\_if\_recursion\_tail\_stop\_-  
     do:nn ..... 09:406,  
     09:425, 07:505, 07:510, 07:519,  
     07:528, 07:574, 07:587, 07:597, 07:641  
 \quark\_new:N ..... 09:15,  
     09:16, 11:40, 07:46, 07:47, 07:1626  
 \q\_recursion\_stop .....  
     ... 09:395, 36:140, 51:20, 07:332,  
     07:343, 07:456, 07:793, 07:1214,  
     07:1447, 07:1709, 07:1724, 07:1726,  
     07:1733, 07:2458, 07:2464, 08:428  
 \q\_recursion\_tail .....  
     ... 09:395, 36:140, 51:19,  
     07:456, 07:793, 07:1214, 07:1447,  
     07:1724, 07:1728, 07:2454, 07:2457,  
     07:2463, 07:2464, 08:427, 08:428, 129  
 \q\_stop ..... 11:79, 11:92, 07:420,  
     07:601, 07:610, 07:615, 07:1300,  
     07:1302, 07:1642, 07:1659, 07:2266

quark internal commands:

\q\_\_cmd .... 07:404, 07:408, 07:994,  
     07:1330, 07:1331, 07:1413, 07:1421,  
     07:1624, 07:2013, 07:2014, 07:2017,  
     07:2022, 07:2028, 07:2029, 07:2031,  
     07:2035, 07:2040, 07:2043, 07:2046,  
     07:2057, 07:2061, 07:2062, 07:2065,  
     07:2066, 07:2069, 07:2080, 07:2083,  
     07:2087, 07:2098, 07:2100, 07:2101,  
     07:2102, 07:2103, 07:2104, 07:2105,  
     07:2106, 07:2110, 07:2129, 07:2130,  
     07:2131, 07:2132, 07:2133, 07:2134,  
     07:2135, 07:2136, 07:2137, 07:2138,  
     07:2139, 07:2140, 07:2141, 07:2142,  
     07:2145, 07:2150, 07:2156, 07:2157,  
     07:2163, 07:2168, 07:2169, 07:2172,  
     07:2183, 07:2190, 07:2195, 07:2196,  
     07:2197, 07:2200, 07:2201, 07:2203, 168  
 \q\_\_cmd\_recursion\_stop .....  
     ... 07:47, 07:2335,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

07:2337, 07:2346, 07:2369, 07:2378, 177  
 \q\_\_cmd\_recursion\_tail . . . . . 07:46, 07:2335, 116  
 \q\_\_hook\_recursion\_stop . . . . . 09:15, 09:326, 09:327, 09:336, 09:361  
 \q\_\_hook\_recursion\_tail . . . . . 09:15, 09:326, 09:340  
 \q\_\_template\_nil . . . 11:40, 11:881, 352  
 \quotedblbase . . . 21:555, 21:783, 21:1176  
 \quotesinglbase . . . . . 21:556, 21:1173

**R**

\r . . . . . 21:236, 21:388, 21:431,  
 21:470, 21:610, 21:637, 21:647,  
 21:673, 21:756, 21:795, 21:1245,  
 21:1263, 21:1289, 21:1411, 21:1412,  
 33:183, 33:205, 02:443, 02:444, 1213  
 \radical .. 28:1104, 28:1107, 28:1137, 1245  
 \raggedbottom . . . . . 49:82  
 \raggedleft . 37:460, 37:478, 37:489, 37:496  
 \raggedright . 37:455, 37:474, 37:488, 37:494  
 \raise . . . . . 21:327, 21:359, 21:430, 21:433,  
 21:732, 21:797, 21:895, 21:1253,  
 29:651, 30:472, 30:520, 30:522,  
 38:73, 40:579, 40:588, 42:61, 42:73,  
 42:105, 42:117, 42:195, 42:305,  
 42:452, 42:494, 42:525, 42:550,  
 42:618, 42:635, 42:636, 42:739, 42:795  
 \raisebox . . . . . 819  
 \raisebox . . . . . 21:872, 21:1222, 40:556  
 \rangle . . . . . 30:599  
 \RawIndent . . . . . 16:136, 16:172, 406  
 \RawNoIndent . . . . . 16:136  
 \RawNoindent . . . . . 16:137, 16:173, 406  
 \RawParEnd . . . . . 16:136, 16:174, 02:482, 406  
 \RawShipout . . . . . 53:151, 53:438, 1065  
 \rbrace . . . . . 21:309, 30:603  
 \rbrack . . . . . 02:441  
 \rceil . . . . . 30:607  
 \Re . . . . . 30:320  
 \read . . . . . 462  
 \ReadonlyShipoutCounter . . . . .  
     . . . . . 53:346, 53:440, 1064  
 \RecordProperties . . . . . 36:80, 36:270, 749  
 \Ref . . . . . 739  
 \Ref . . . . . 35:112, 35:149, 35:151, 35:170,  
     35:178, 35:182, 35:189, 35:193, 35:200  
 \ref . . . . . 35:10, 35:178, 35:189, 45:553, 746  
 \RefProperty . . . . . 36:118, 36:271, 748  
 \refstepcounter . . . . . 512  
 refstepcounter (socket) . . . . . 35:117  
 \refstepcounter . . . . . 35:111, 35:172, 35:182,  
     35:184, 35:193, 35:195, 38:350,  
     38:507, 39:226, 43:37, 44:59, 45:9, 1164

refstepcounter/target (socket) . . . . . 35:118  
 \RefUndefinedWarn . . . . . 36:214, 36:272, 748  
 regex commands:  
     \regex\_match:nnTF . . . . . 11:383  
 \registernumber . . . . . 44  
 registernumber . . . . . 04:389  
 \relax . . . . . 1217  
 \relax\_ . . . . . 1211  
 \Relbar . . . . . 30:483, 30:491, 30:493, 30:499, 1239  
 \relbar . . . . . 30:480, 30:495, 30:497  
 \relpenalty . . . . . 02:324  
 remove commands:  
     remove\_from\_callback . . . . . 04:907  
 \remove\_from\_callback . . . . . 46  
 \RemoveFromHook . . . . .  
     . . . . . 08:2770, 08:2899, 37:39, 37:40,  
     37:41, 37:104, 37:105, 37:106, 198  
 \removelastskip . . . . .  
     . . . . . 02:514, 02:516, 02:518, 02:520  
 \renewcommand . . . . . 81  
 \renewcommand . . . . . 30:74, 30:76, 30:78,  
     30:79, 30:81, 30:83, 30:85, 30:86,  
     30:92, 30:94, 30:96, 30:97, 30:110,  
     30:111, 30:112, 30:120, 30:121,  
     38:455, 38:475, 38:496, 06:124, 98  
 \RenewCommandCopy . . . . . 06:473, 06:475, 98  
 \RenewDocumentCommand . . . . . 07:2804  
 \RenewDocumentEnvironment . . . . .  
     . . . . . 07:2602, 07:2840  
 \renewenvironment . . . . . 82  
 \renewenvironment . . . . . 38:504, 38:516, 06:152  
 \RenewEnvironmentCopy . 06:514, 06:516, 1264  
 \RenewExpandableDocumentCommand . . . . .  
     . . . . . 48:484,  
     48:485, 48:486, 48:487, 07:2923, 190  
 \repeat . . . . . 04:154, 04:164, 24:714,  
     41:382, 50:1246, 50:1307, 50:1377,  
     50:1440, 50:1491, 50:1529, 57:369,  
     57:380, 57:390, 57:401, 57:431,  
     57:457, 57:467, 01:65, 01:67, 02:490  
 \requestedLaTeXdate . . . . . 08:2919,  
     50:1592, 50:1632, 50:1652, 50:1739  
 \requestedpatchdate . . . . . 50:1662, 50:1740  
 \RequirePackage . . . . . 976  
 \RequirePackage . 04:24, 50:676, 50:683,  
     50:726, 50:735, 50:1189, 54:2005, 1065  
 \RequirePackageWithOptions . . . . . 976  
 \RequirePackageWithOptions . . . . . 50:718  
 reserved@**a** commands:  
     \reserved@a: . . . . . 20:233,  
     20:305, 20:367, 20:414, 50:1219, 50:1354  
 reserved@**b** commands:  
     \reserved@b: . . . . . 50:232, 50:249

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

reserved@c commands:

- \reserved@c ..... 20:755
- \RestoreAtCatcode ..... 1209
- \restorecr ..... 18:556
- \ResumeTagging ..... 1161
- \ResumeTagging ..... 23:22, 55:3, 1161
- \ReverseBoolean ..... 07:3001
- \reversemarginpar ..... 45:387
- \rfloor ..... 30:611
- \rgroup ..... 30:615
- \rhd ..... 29:733
- \rho ..... 30:290
- \rhook ..... 30:486, 30:487
- \right ..... 30:632, 30:634, 30:636, 30:638, 30:643, 30:644, 30:645, 30:646, 38:168, 38:173, 38:197
- \Rightarrow ..... 30:418, 30:493, 30:505
- \rightarrow ..... 30:445, 30:447, 30:451, 30:485, 30:495, 30:503, 30:556
- \rightarrowfill ..... 30:538, 30:554
- \rightharpoondown ..... 30:462
- \rightharpoonup ..... 30:461, 30:473
- \righthyphenmin ..... 56:11, 02:371
- \rightleftharpoons ..... 30:471
- \rightline ..... 40:614
- \rightmargin ..... 39:9, 39:40, 39:51, 803
- \rightmark ..... 49:77, 947
- \rightskip ..... 24:685, 37:452, 37:456, 37:462, 37:472, 37:475, 37:480, 37:529, 37:551, 39:75, 40:392, 40:413, 44:207, 44:230, 02:417, 02:528
- \rlap ..... 21:430, 21:433, 21:797, 38:428, 38:456, 40:618, 41:81, 1225
- \rm ..... 1201
- \rmdefault ..... 29:6, 29:208, 29:398, 29:407, 29:439, 29:450, 29:482, 30:58, 30:127, 33:7, 33:613, 648
- \rmfamily .. 29:4, 29:5, 29:405, 29:448, 29:449, 29:480, 29:481, 32:15, 1226
- rmfamily ..... 29:420
- \rmmath ..... 1194
- \rmoustache ..... 30:572
- \rmsubstdefault ..... 30:18, 30:30, 33:28, 33:39, 33:85
- \Roman ..... 512
- \Roman ..... 22:186, 1064
- \roman ..... 512
- \roman ..... 22:185
- \romannumeral ..... 22:191, 22:192, 37:316, 37:348, 37:365, 39:43, 39:258, 39:269, 771
- \root ..... 38:66, 38:356
- \rootbox ..... 38:66
- \rq ..... 02:439
- \rule ..... 819
- \rule ..... 40:490, 40:507, 40:524, 40:530, 45:477, 45:495, 45:513

**S**

- \S ..... 21:311, 1218
- \safesubencodingfoundfalse ..... 33:1255
- \safesubencodingfoundtrue ..... 33:1245
- \samepage ..... 421
- \samepage ..... 18:11, 18:29, 1195
- \SaveAtCatcode ..... 1209
- \savebox ..... 818
- \savebox ..... 40:155
- \savecatcodetable .. 04:117, 04:168, 04:170
- \savepos ..... 751
- \sb ..... 38:212
- \sbox ..... 818
- \sbox ..... 19:4, 21:498, 21:514, 39:229, 40:161, 40:168, 40:172, 40:177, 40:182, 02:522, 1216
- \sc ..... 566

scan commands:

- \scan\_new:N ..... 11:38, 11:39, 08:44
- \scan\_stop: 08:2530, 08:2531, 09:452, 09:453, 09:511, 09:512, 09:547, 09:548, 16:94, 48:104, 48:106, 48:108, 48:109, 48:251, 48:252, 48:253, 51:85, 51:97, 53:44, 53:325, 53:327, 53:328, 07:1856, 07:1857, 08:661, 08:668, 08:679, 08:688, 08:723, 08:734, 08:770, 08:782, 08:792, 08:822, 08:854, 08:1196, 412

scan internal commands:

- \s\_file\_stop ..... 52:103, 52:105, 52:108, 52:110, 52:112, 52:125
- \s\_hook\_mark ..... 08:2247, 08:2261, 08:2310, 08:2318, 08:2421, 08:2423, 08:2484, 08:2486, 08:2490, 08:2491, 08:2498, 08:2499, 08:2512, 08:2513, 09:24, 09:26, 09:35, 09:37, 09:113, 09:118, 09:312, 09:323, 09:457, 09:460, 09:472, 09:476, 09:481, 09:516, 09:536, 09:552, 09:564, 08:44, 08:45, 08:46, 08:348, 08:351, 08:354, 08:358, 08:361, 08:362, 08:712, 08:763, 08:765, 08:773, 08:775, 08:778, 08:780, 08:882, 08:907, 08:908, 08:912, 08:1217, 08:1234, 08:1238, 08:1280, 08:1281, 08:1336, 224
- \s\_keys\_stop ... 51:47, 51:175, 51:176
- \s\_template\_mark .. 11:38, 11:881, 352
- \s\_template\_stop ..... 11:39, 11:286, 11:297, 11:308, 11:329,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

```

11:343, 11:351, 11:430, 11:433,
11:740, 11:743, 11:881, 11:885, 352
\scdefault 25:694, 29:27, 30:102
\scriptfont 26:338
\scriptscriptfont 26:339
\scriptscriptstyle 38:65, 38:68
\scriptspace 02:399
\scriptstyle 30:344, 38:64
\scshape 21:300, 25:692,
 25:693, 29:25, 29:26, 32:23, 1235
\searrow 30:413
\sec 38:20
\secdef 44:142
\secondoftwo 457
\section 304
\sectionmark 44:143, 946
\selectfont 19:7, 21:302, 21:329, 21:360,
 21:449, 21:809, 21:871, 21:1221,
 21:1255, 21:1573, 24:291, 24:301,
 24:311, 25:528, 25:533, 25:538,
 25:684, 25:688, 25:691, 25:694,
 25:697, 25:700, 25:703, 25:706,
 26:114, 26:115, 26:159, 26:162,
 26:164, 29:6, 29:9, 29:12, 29:15,
 29:18, 29:21, 29:24, 29:27, 29:30,
 29:264, 29:287, 29:325, 29:345,
 29:360, 29:371, 29:382, 29:385,
 29:409, 29:414, 29:419, 29:452,
 29:457, 29:462, 29:476, 29:479,
 29:482, 29:485, 29:488, 29:564,
 29:641, 29:665, 29:682, 29:697,
 33:36, 33:94, 33:104, 33:636, 33:673,
 33:913, 33:930, 45:403, 45:424, 222
selectfont 26:150
seq commands:
 \seq_clear:N 11:160, 11:211, 08:1557, 08:1639
 \seq_clear_new:N 08:1565, 08:1647
 \seq_const_from_clist:Nn 11:16
 \seq_count:N 55:130
 \seq_gclear_new:N 11:115
 \seq_gpop:NN 08:2561
 \seq_gpop:NNTF 52:72, 08:451, 08:459
 \seq_gpop_right:NN 08:430
 \seq_gpush:Nn 08:2549, 08:2555, 52:64, 08:444
 \seq_gput_right:Nn 48:23, 08:108, 08:137, 08:419, 08:423
 \seq_gset_eq:NN 11:116
 \seq_if_empty:NTF 08:418, 08:469
 \seq_if_exist:NTF 11:155, 52:60
 \seq_if_in:NnTF 11:230, 48:9, 48:218
 \seq_map_break: 11:249, 11:340
 \seq_map_function:NN 11:226, 11:346, 11:707
 \seq_map_inline:Nn 08:2942, 48:124,
 48:157, 48:195, 48:206, 48:354,
 48:430, 07:2227, 08:1426, 08:1453,
 08:1581, 08:1599, 08:1664, 08:1687
 \seq_mapthread_function:NNN ... 180
 \seq_new:N 08:2539, 10:36, 11:29, 48:6, 52:61,
 55:133, 07:2211, 08:28, 08:33, 08:1548
 \seq_put_right:Nn .. 10:86, 11:263,
 08:1563, 08:1645, 08:1748, 08:1755
 \seq_set_eq:NN 11:157
 \seq_set_split:Nnn ... 55:129, 07:2225
 \seq_use:Nnnn .. 10:60, 08:1808, 08:1813
seq internal commands:
 \g_mark_classes_seq 48:6,
 48:9, 48:23, 48:124, 48:157, 48:195,
 48:206, 48:218, 48:354, 48:430, 950
 \series 1194
 \seriesdefault 21:1574, 28:390,
 29:220, 29:222, 29:660, 29:678,
 29:694, 29:711, 29:773, 30:127, 642
 \setattribute 43
 \setattribute 04:82, 04:239
 \setbox 812
 \setbox... 1233
 \setbox0 1229
 \setcounter 512
 \setcounter 20:445,
 22:2, 22:42, 22:62, 28:143, 39:249,
 54:2328, 54:2331, 54:2334, 54:2338, 607
 \SetDefaultHookLabel 08:2772, 200
 \SetKeys 20:751, 51:255, 1028
 \setlength 521
 \setlength 18:84,
 18:244, 18:259, 18:518, 23:4, 38:524,
 38:529, 38:530, 38:531, 40:47, 40:62,
 40:247, 40:266, 40:333, 40:336,
 40:360, 40:363, 40:439, 40:546,
 40:547, 40:548, 40:577, 40:578,
 40:585, 40:586, 40:587, 41:176,
 41:384, 54:2344, 54:2345, 54:2346,
 54:2349, 54:2350, 54:2354, 54:2355,
 54:2356, 54:2360, 54:2361, 54:2362, 426
 \SetMathAlphabet 24:11,
 27:140, 27:141, 28:695, 30:162, 30:163
 \setminus 30:400
 \SetProperty 36:38, 36:268, 749
 \setrangepagecode 04:96, 04:104, 04:113, 04:114
 \SetSymbolFont 28:530, 30:152, 30:153, 30:154, 1198
 \SetTemplateKeys 11:1194, 349

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\settodepth 521
\settodepth 23:17
\settoheight 521
\settoheight 23:17
\settowidth 521
\settowidth 23:17
\sf 1201
\sfcodes 18:463, 20:45,
20:115, 20:173, 57:250, 57:545,
02:431, 02:432, 02:433, 02:434, 02:551
\sfddefault 29:9, 29:212, 29:399,
29:412, 29:440, 29:455, 29:485, 30:58
\sffamily .. 29:7, 29:8, 29:410, 29:453,
29:454, 29:483, 29:484, 32:16, 635
\sffamily 29:420
\sfsubstdefault 30:19, 30:31, 33:30, 33:87
\shape 1196
\shapedefault
.. 21:1574, 25:684, 28:391, 29:661,
29:679, 29:695, 29:712, 30:127, 642
\sharp 30:337
\shipout 53:4, 53:52,
53:431, 53:434, 54:606, 54:664, 1215
shipout 53:153
shipout commands:
\l_shipout_box
. 53:23, 53:35, 53:38, 53:50, 53:61,
53:126, 53:149, 53:179, 53:184,
53:207, 53:208, 53:215, 53:226,
53:229, 53:230, 53:234, 53:241,
53:251, 53:259, 53:266, 53:276,
53:282, 53:283, 53:286, 53:293,
53:295, 53:296, 53:302, 53:304, 1060
\l_shipout_box_dp_dim
..... 53:194, 53:197,
53:199, 53:230, 53:283, 53:512, 1060
\l_shipout_box_ht_dim
..... 53:193, 53:197, 53:199,
53:229, 53:249, 53:282, 53:511, 1060
\l_shipout_box_ht_plus_dp_dim ...
..... 53:196, 53:199,
53:215, 53:266, 53:277, 53:279, 1060
\l_shipout_box_wd_dim ... 53:195,
53:199, 53:241, 53:293, 53:513, 1060
\shipout_debug_off:
..... 53:7, 53:13, 53:415, 1064
\shipout_debug_on:
..... 53:7, 53:8, 53:414, 1064
\shipout_discard:
..... 53:343, 53:343, 53:411, 1063
\g_shipout_READONLY_int
..... 36:233, 53:102, 53:104,
53:111, 53:116, 53:346, 53:355,
53:359, 53:363, 53:367, 53:371, 1064
\g_shipout_totalpage_int 1064
\g_shipout_totalpages_int
..... 53:87, 53:348, 1064
shipout internal commands:
__shipout_add_background_box:n .
53:180, 53:206, 53:206, 53:338, 53:420
__shipout_add_background_-
picture:n
..... 53:69, 53:337, 53:337, 53:424
__shipout_add_firstpage_-
material:Nn
53:173, 53:189, 53:189, 53:413, 53:418
__shipout_add_firstpage_-
specials:
53:110,
53:166, 53:178, 53:178, 53:181, 1071
__shipout_add_foreground_box:n .
53:117, 53:257, 53:257, 53:341, 53:422
__shipout_add_foreground_-
picture:n
..... 53:64, 53:340, 53:340, 53:426
__shipout_debug:n 53:7, 53:7,
53:20, 53:103, 53:115, 53:148, 1067
\g__shipout_debug_bool
..... 53:6, 53:10, 53:15, 53:21
__shipout_debug_gset:
..... 53:7, 53:11, 53:16, 53:18
\g__shipout_discard_bool
.... 53:81, 53:88, 53:90, 53:203, 53:344
__shipout_drop_firstpage_-
specials:
53:127,
53:167, 53:178, 53:183, 53:185, 1074
__shipout_excuse_extra_page: ...
..... 53:384, 53:392, 53:392
__shipout_execute:
..... 53:46, 53:46, 53:52, 1068
__shipout_execute_cont:
..... 53:57, 53:59, 53:59
__shipout_execute_main_cont:Nnnn
.... 53:60, 53:77, 53:77, 53:147, 1072
__shipout_execute_nohooks_cont:
..... 53:144, 53:146
__shipout_execute_raw:
..... 53:135, 53:135, 53:151, 1072
__shipout_execute_test_level: ...
..... 53:49, 53:54, 53:54
__shipout_execute_test_level_-
raw:
..... 53:135, 53:138, 53:141
__shipout_finalize_box:
..... 53:26, 53:28, 53:44, 53:124
\l__shipout_firstpage_box
..... 53:170, 53:180, 53:187, 1074
__shipout_get_box_size:N . 53:85,
53:107, 53:192, 53:192, 53:207, 1075

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\l__shipout_group_level_tl
 ... 53:47, 53:53, 53:56, 53:136, 53:143
\c__shipout_horigin_t1 53:310, 53:325
__shipout_init_page_origins: ...
 ... 53:310, 53:310, 53:321, 53:324
\g__shipout_lastpage_handled_-
 bool
 53:120, 53:188, 53:365
__shipout_picture_overlay:n ...
 ... 53:323, 53:323, 53:338, 53:341
\l__shipout_raw_box 53:25, 53:139,
 53:147, 53:149, 53:179, 53:184, 1071
__shipout_run_firstpage_hook: ...
 . 53:108, 53:163, 53:163, 53:172, 1071
\l__shipout_saved_badness_t1 ...
 ... 53:204, 53:210, 53:218, 53:231,
 53:236, 53:244, 53:253, 53:261,
 53:269, 53:281, 53:288, 53:298, 53:306
__shipout_saved_protect: ...
 ... 53:83, 53:132, 53:152, 53:152
\l__shipout_tmp_box
 ... 53:204, 53:217, 53:219, 53:220,
 53:221, 53:225, 53:243, 53:245,
 53:246, 53:247, 53:250, 53:268,
 53:270, 53:271, 53:272, 53:278,
 53:297, 53:299, 53:300, 53:301,
 53:303, 53:329, 53:331, 53:332, 53:333
\c__shipout_vorigin_t1 53:310, 53:327
shipout/after 53:153, 1060
shipout/background 53:153, 1060
shipout/before 53:153, 1060
shipout/firstpage 53:153, 1060
shipout/foreground 53:153, 1060
shipout/lastpage 53:153, 1060
\ShipoutBox ... 53:23, 53:439, 53:494, 1060
\ShipoutBoxDepth 53:459, 53:511
\ShipoutBoxHeight ... 53:458, 53:511, 1086
\ShipoutBoxWidth 53:460, 53:511
\shortstack
 42:132, 42:147, 42:152, 42:814, 42:831
\show 06:552, 06:625, 100
show commands:
 \show_hook:n 275
 \showbox 54:1919
 \showboxbreadth 02:631, 02:655,
 02:672, 02:697, 54:1919, 02:387, 02:554
 \showboxdepth 02:630,
 02:654, 02:671, 02:698, 24:672,
 24:717, 24:734, 54:1919, 02:388, 02:554
 \ShowCommand . 06:545, 07:1311, 07:1340,
 07:1352, 07:1360, 07:1363, 07:1524, 100
 \ShowEnvironment 06:680, 104
 \ShowFloat 54:1900
 \ShowHook 08:2796, 08:2904, 205
 \showhyphens 24:661, 1268
 \ShowInstanceValues 11:1176, 349
 \ShowMarksAt 48:358
 \showoutput 02:553, 1226
 \showoverfull 02:577,
 02:612, 02:620, 02:552, 02:555, 1226
 \ShowSocket ... 10:149, 10:170, 10:183, 335
 \ShowTemplateCode 11:1176, 350
 \ShowTemplateDefaults 11:1176, 350
 \ShowTemplateInterface 11:1176, 350
 \ShowTemplateVariables 11:1176, 350
 \showtokens 06:660, 105
 \Sigma 30:310
 \sigma 30:291
 \sim 30:452, 30:464
 \simeq 30:453
 \sin 38:9
 \sinh 38:11
 \size 1194
 \skew 30:551
 \skip 04:33, 40:464,
 45:391, 54:318, 54:494, 02:28,
 02:53, 02:92, 02:209, 02:250, 02:295
skip commands:
 \skip_eval:n 11:370, 05:165
 \skip_new:N 11:36
 \skip_set:Nn 16:25
 \skip_zero:N 16:58, 53:222,
 53:223, 53:224, 53:273, 53:274, 53:275
 \l_tmpa_skip 346
 \c_zero_skip 16:26
 \skipdef 04:229, 02:45, 02:53, 02:92
 \skipeval 77
 \skipeval 05:158, 05:175, 77
 \skipzero 04:229
 \slash 06:890, 06:911, 02:507
 \sldefault 25:691, 29:24, 30:102
 \SLiTeX 1216
 \sloppy 40:397, 40:416, 49:86, 49:91
 \sloppypar (env.) 49:91
 \sloppypar 49:91
 \slshape 21:440, 21:800, 25:689,
 25:690, 29:22, 29:23, 32:22, 33:627
 \small 198
 \smallbreak 06:891, 06:912, 02:515
 \smallint 30:364
 \smallskip 18:447, 02:516
 \smallskipamount .. 18:447, 18:450, 02:515
 \smash 17:29, 17:51, 30:480, 30:554,
 30:555, 30:558, 30:559, 38:126, 788
 \smile 30:457
socket commands:
 \socket_assign_plug:nn 10:42,
 10:44, 10:116, 10:116, 10:174, 334

```

**File Key:** 01=ltexprchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\socket_debug_off: 10:7, 10:14, 10:177, 335
\socket_debug_on: 10:7, 10:9, 10:176, 335
\socket_log:n 10:53, 10:53, 10:72, 10:172, 335
\socket_new:nn 10:27, 10:27, 10:170, 334
\socket_new_plugin:nnn 10:38, 10:41, 10:73, 10:73, 10:173, 334
\socket_set_plugin:nnn 10:73, 10:95, 10:163, 334
\socket_show:n 10:53, 10:72, 10:171, 335
\socket_use:n 10:132, 10:138, 335
\socket_use:nn 10:132, 10:139, 335
\socket_use:nnn .. 10:132, 10:140, 335
\socket_use:nnnn .. 10:132, 10:141, 335
\socket_use:nw 10:132, 10:132, 10:138, 10:139, 10:140, 10:141, 10:175, 339
\socket_use_expandable:n 10:142, 10:145, 335
\socket_use_expandable:nw 10:142, 10:142, 10:145, 335
socket internal commands:
__socket_debug:n 10:7, 10:7, 10:21, 336
\g__socket_debug_bool 10:6, 10:11, 10:16, 10:22, 10:24
__socket_debug_gset: 10:7, 10:12, 10:17, 10:19
__socket_debug_term:n 10:7, 10:8, 10:23, 10:45, 10:87, 10:105, 10:121, 10:133, 336
Sockets:
refstepcounter 35:117
refstepcounter/target 35:118
tagsupport/caption/begin 55:94
tagsupport/caption/end 55:94
tagsupport/caption/label/begin 55:96
tagsupport/caption/label/end . 55:96
tagsupport/float/begin 55:92
tagsupport/float/end 55:92
tagsupport/float/hmode/begin . 55:90
tagsupport/float/hmode/end ... 55:90
tagsupport/para/begin 55:49
tagsupport/para/end 55:49
tagsupport/para/restore 55:39
tagsupport/refstepcounter 55:49
tagsupport/tbl/cell/begin 55:69
tagsupport/tbl/cell/end 55:69
tagsupport/tbl/colspan 55:79
tagsupport/tbl/finalize 55:77
tagsupport/tbl/hmode/begin ... 55:80
tagsupport/tbl/hmode/end 55:80
tagsupport/tbl/init 55:75
tagsupport/tbl/init/celldata . 55:76
tagsupport/tbl/leaders/begin . 55:88
tagsupport/tbl/leaders/end ... 55:88
tagsupport/tbl/longtable/finalize 55:84
tagsupport/tbl/longtable/foot . 55:86
tagsupport/tbl/longtable/head . 55:86
tagsupport/tbl/longtable/init . 55:84
tagsupport/tbl/pcell/end 55:69, 55:69
tagsupport/tbl/restore/celldata 55:78
tagsupport/tbl/row/begin 55:69
tagsupport/tbl/row/end 55:69
tagsupport/tbl/vmode/begin ... 55:80
tagsupport/tbl/vmode/end 55:80
tagsupport/toc/contentsline/after 55:63
tagsupport/toc/contentsline/before 55:63
tagsupport/toc/leaders/after . 55:67
tagsupport/toc/leaders/before . 55:67
tagsupport/toc/starttoc/after . 55:65
tagsupport/toc/starttoc/before 55:65
\sourceLaTeXdate 08:2918, 03:164, 50:65, 50:103
\sp 38:212
\space 02:447, 1212
\spacefactor 17:28, 17:30, 17:50, 17:52, 18:130, 18:139, 18:158, 18:172, 18:184, 18:198, 18:212, 18:463, 18:504, 18:509, 21:70, 21:73, 45:531, 45:533, 02:505, 02:506, 1237
\spacefactor_in_math_mode_gh/643 . 1260
\spaceskip 33:6, 33:612, 02:421
\spadesuit 30:341
\span 41:383
\special 17:26, 17:48, 1061
\SplitArgument 07:3001
\splitfirstmark 54:2249
\SplitList 07:3001, 192
\splitmaxdepth 45:469, 45:488, 45:506, 54:2243, 02:394
\splittopskip 45:468, 45:487, 45:505, 02:419
\sqcap 30:383
\sqcup 30:384
\sqrt 38:355, 1225
\sqrtsign 30:536, 38:71, 38:355, 1225
\sqsubset 29:729
\sqsubseteqq 30:406
\sqsupset 29:730
\sqsupseteqq 30:407
\SS .. 21:304, 21:534, 21:1157, 57:650, 1230
\ss 21:254, 21:406, 21:557, 21:784, 21:1132, 57:650
\sscdefault 25:536, 25:609, 25:706

```

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspace.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

```

\sscsshape 25:536, 25:608, 25:704, 25:705, 32:31
\stackrel 38:353
\stamp 1204
\star 30:404
\stepcounter 512
\stepcounter 22:17, 22:27, 24:626, 28:48, 35:124, 35:134, 35:141, 35:172, 35:184, 35:195, 38:363, 38:423, 38:517, 45:452, 45:520, 54:650, 54:709, 1197
\stockheight 54:95
\stockwidth 54:95
\stop 33:1636, 37:240
\storedpar 04:156, 04:161
str internal commands:
 \c_backslash_str 09:147, 09:398, 07:2554
 \c_hash_str 09:481, 09:485, 09:509, 07:1384, 07:1510
 \c_right_brace_str 08:2056
 \str_case:nn 09:595, 11:510, 57:562, 07:1119
 \str_case:nnTF 09:40, 11:448, 36:29, 51:30, 07:2789
 \str_case_e:nnTF 07:2475
 \str_count:n 09:161, 09:236, 08:37, 08:43, 08:1249, 08:1853
 \str_gset:Nn 08:2781
 \str_head:n 07:2420
 \str_if_empty:NTF 57:584
 \str_if_eq:nn 263
 \str_if_eq:NNTF 52:555
 \str_if_eq:nnTF 08:2116, 08:2312, 08:2516, 08:2570, 08:2640, 09:215, 09:287, 09:369, 09:520, 09:527, 09:556, 09:559, 11:79, 11:92, 11:243, 11:264, 11:462, 11:477, 11:493, 11:565, 11:722, 11:724, 11:898, 05:137, 52:131, 52:549, 52:559, 07:100, 07:102, 07:425, 07:1198, 07:1203, 07:1615, 07:2006, 07:2019, 07:2021, 07:2051, 07:2071, 07:2073, 07:2092, 07:2119, 07:2147, 07:2149, 07:2174, 07:2176, 07:2199, 07:2316, 07:2328, 07:2356, 07:2532, 08:346, 08:438, 08:479, 08:535, 08:537, 08:608, 08:610, 08:740, 08:798, 08:831, 08:884, 08:966, 08:973, 08:1001, 08:1008, 08:1801, 08:1909, 08:2009, 168
 \str_if_eq_p:nn .. 07:485, 07:486, 07:487, 07:488, 07:935, 07:2364, 07:2386, 08:888, 08:1592, 08:1678
\str_if_exist:NTF 10:28, 10:55, 10:74, 10:96, 10:117
\str_lowercase:n 57:596
\str_map_function:NN 05:144
\str_new:N 10:35, 07:19
\str_replace_all:Nnm 09:508
\str_set:Nn 10:124, 52:399, 52:400, 57:582, 57:586, 07:228, 07:242, 07:243, 07:287
\str_tail:n 07:2006, 07:2417, 07:2443, 168
\str_use:N 10:62, 10:65, 10:122, 10:135, 10:136, 10:143

```

str internal commands:

```

 __str_if_eq:nn 08:23, 223
\strcmp 50:346, 50:362
\stretch 18:532
\string 1232
\stripmeaning 1201
\strut 38:191, 38:192, 41:29, 06:892, 06:913, 02:523, 222
\strutbox 26:189, 40:490, 40:507, 40:524, 41:186, 41:187, 45:469, 45:477, 45:488, 45:495, 45:506, 45:513, 02:523
\subencodingresult .. 33:1236, 33:1241, 33:1410, 33:1414, 33:1416, 33:1417
\subparagraphmark 44:143
\subsectionmark 44:143, 946
\subset 30:433
\subsepeq 30:435
\substdefault 1250
\subsubsectionmark 44:143
\succ 30:427
\suceed 30:430
\sum 30:357
\sup 38:24
\supereject 1228
\suppressfloats 54:2010
\superset 30:432
\supsepeq 30:434
\surd 30:343
\SuspendTagging 1161
\SuspendTagging 23:22, 55:3, 1161
\swallow 30:415
\swdefault 25:531, 25:607, 25:703
\swshape 25:531, 25:606, 25:701, 25:702, 32:30
\symbol 21:162, 29:619
\symletters ... 33:8, 33:14, 33:614, 33:1136
\symoperators 30:650
sys commands:
 \sys_if_engine_luatex:TF 53:26

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

| T                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \T . . . . .                                          | <a href="#">14:23</a> , <a href="#">21:335</a> ,<br><a href="#">21:337</a> , <a href="#">21:339</a> , <a href="#">21:341</a> , <a href="#">21:343</a> ,<br><a href="#">21:345</a> , <a href="#">21:347</a> , <a href="#">21:349</a> , <a href="#">21:351</a> ,<br><a href="#">21:374</a> , <a href="#">50:1572</a> , <a href="#">50:1576</a> , <a href="#">50:1577</a>                                                                           |
| \t . . . . .                                          | <a href="#">21:282</a> , <a href="#">21:741</a> , <a href="#">21:849</a> , <a href="#">33:157</a> ,<br><a href="#">33:158</a> , <a href="#">33:185</a> , <a href="#">33:189</a> , <a href="#">33:191</a> ,<br><a href="#">33:203</a> , <a href="#">33:206</a> , <a href="#">33:208</a> , <a href="#">33:688</a> ,<br><a href="#">33:856</a> , <a href="#">33:1123</a> , <a href="#">33:1125</a> , <a href="#">33:1318</a> , <a href="#">1218</a> |
| tabbing (env.) . . . . .                              | <a href="#">41:71</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tabbing . . . . .                                    | <a href="#">41:71</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tabbingsep . . . . .                                 | <a href="#">41:130</a> , <a href="#">41:132</a> , <a href="#">41:166</a>                                                                                                                                                                                                                                                                                                                                                                         |
| \tabcolsep . . . . .                                  | <a href="#">41:259</a> , <a href="#">41:338</a>                                                                                                                                                                                                                                                                                                                                                                                                  |
| \tableofcontents . . . . .                            | <a href="#">766</a>                                                                                                                                                                                                                                                                                                                                                                                                                              |
| \tabskip . . . . .                                    | <a href="#">38:208</a> , <a href="#">38:209</a> ,<br><a href="#">38:369</a> , <a href="#">38:372</a> , <a href="#">38:375</a> , <a href="#">38:377</a> ,<br><a href="#">38:522</a> , <a href="#">38:535</a> , <a href="#">38:538</a> , <a href="#">38:540</a> ,<br><a href="#">41:167</a> , <a href="#">41:192</a> , <a href="#">02:420</a> , <a href="#">02:534</a> , <a href="#">1198</a>                                                      |
| tabular (env.) . . . . .                              | <a href="#">41:174</a>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| \tabular . . . . .                                    | <a href="#">41:174</a>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| \tabular* . . . . .                                   | <a href="#">41:175</a>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| \tabularnewline . . . . .                             | <a href="#">41:194</a> , <a href="#">41:207</a> , <a href="#">1222</a>                                                                                                                                                                                                                                                                                                                                                                           |
| tag commands:                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \tag_get:n . . . . .                                  | <a href="#">55:58</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tag_if_active: . . . . .                             | <a href="#">55:10</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tag_if_active:TF . . . . .                           | <a href="#">55:8</a> , <a href="#">55:135</a>                                                                                                                                                                                                                                                                                                                                                                                                    |
| \tag_resume:n . . . . .                               | <a href="#">55:3</a> , <a href="#">55:4</a> , <a href="#">55:6</a>                                                                                                                                                                                                                                                                                                                                                                               |
| \tag_socket_use:n . . . . .                           | <a href="#">55:7</a> , <a href="#">55:15</a> , <a href="#">1162</a>                                                                                                                                                                                                                                                                                                                                                                              |
| \tag_socket_use:nn . . . . .                          | <a href="#">55:7</a> , <a href="#">55:16</a> , <a href="#">1162</a>                                                                                                                                                                                                                                                                                                                                                                              |
| \tag_socket_use:nnn . . . . .                         | <a href="#">55:7</a> , <a href="#">55:17</a> , <a href="#">1162</a>                                                                                                                                                                                                                                                                                                                                                                              |
| \tag_socket_use_expandable:n . . . . .                | <a href="#">55:7</a> , <a href="#">55:14</a> , <a href="#">1162</a>                                                                                                                                                                                                                                                                                                                                                                              |
| \tag_suspend:n . . . . .                              | <a href="#">55:3</a> , <a href="#">55:3</a> , <a href="#">55:5</a>                                                                                                                                                                                                                                                                                                                                                                               |
| tag internal commands:                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \l__tag_block_flattened_level_- int . . . . .         | <a href="#">55:38</a> , <a href="#">55:45</a>                                                                                                                                                                                                                                                                                                                                                                                                    |
| \l__tag_para_bool . . . . .                           | <a href="#">55:46</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \l__tag_para_flattened_bool . . . . .                 | <a href="#">55:44</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \l__tag_para_main_tag_tl . . . . .                    | <a href="#">55:42</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \l__tag_para_tag_default_tl . . . . .                 | <a href="#">55:43</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \l__tag_para_tag_tl . . . . .                         | <a href="#">55:43</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \g__tag_struct_dest_num_prop . . . . .                | <a href="#">55:56</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tag_resume:n . . . . .                               | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \tag_socket_use:n . . . . .                           | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \tag_socket_use:nn . . . . .                          | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \tag_socket_use:nnn . . . . .                         | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \tag_socket_use_expandable:n . . . . .                | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| \tag_suspend:n . . . . .                              | <a href="#">1161</a>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| tagsupport/caption	begin (socket) . . . . .           | <a href="#">55:94</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/caption/end (socket) . . . . .             | <a href="#">55:94</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/caption/label/begin (socket) . . . . .     | <a href="#">55:96</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/caption/label/end (socket) . . . . .       | <a href="#">55:96</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/float/begin (socket) . . . . .             | <a href="#">55:92</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/float/end (socket) . . . . .               | <a href="#">55:92</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/float/hmode/begin (socket) . . . . .       | <a href="#">55:90</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/float/hmode/end (socket) . . . . .         | <a href="#">55:90</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/para/begin (socket) . . . . .              | <a href="#">55:49</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/para/end (socket) . . . . .                | <a href="#">55:49</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/para/restore (socket) . . . . .            | <a href="#">55:39</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/refstepcounter (socket) . . . . .          | <a href="#">55:49</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/cell/begin (socket) . . . . .          | <a href="#">55:69</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/cell/end (socket) . . . . .            | <a href="#">55:69</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/colspan (socket) . . . . .             | <a href="#">55:79</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/finalize (socket) . . . . .            | <a href="#">55:77</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/hmode/begin (socket) . . . . .         | <a href="#">55:80</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/hmode/end (socket) . . . . .           | <a href="#">55:80</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/init (socket) . . . . .                | <a href="#">55:75</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/init/celldata (socket) . . . . .       | <a href="#">55:76</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/leaders/begin (socket) . . . . .       | <a href="#">55:88</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/leaders/end (socket) . . . . .         | <a href="#">55:88</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/longtable/finalize (socket) . . . . .  | <a href="#">55:84</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/longtable/foot (socket) . . . . .      | <a href="#">55:86</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/longtable/head (socket) . . . . .      | <a href="#">55:86</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/longtable/init (socket) . . . . .      | <a href="#">55:84</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/pcell/end (socket) . . . . .           | <a href="#">55:69</a> , <a href="#">55:69</a>                                                                                                                                                                                                                                                                                                                                                                                                    |
| tagsupport/tbl/restore/celldata (socket) . . . . .    | <a href="#">55:78</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/row/begin (socket) . . . . .           | <a href="#">55:69</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/row/end (socket) . . . . .             | <a href="#">55:69</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/vmode/begin (socket) . . . . .         | <a href="#">55:80</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/tbl/vmode/end (socket) . . . . .           | <a href="#">55:80</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/contentsline/after (socket) . . . . .  | <a href="#">55:63</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/contentsline/before (socket) . . . . . | <a href="#">55:63</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/leaders/after (socket) . . . . .       | <a href="#">55:67</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/leaders/before (socket) . . . . .      | <a href="#">55:67</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/starttoc/after (socket) . . . . .      | <a href="#">55:65</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| tagsupport/toc/starttoc/before (socket) . . . . .     | <a href="#">55:65</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tan . . . . .                                        | <a href="#">38:15</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| \tanh . . . . .                                       | <a href="#">38:17</a>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| target . . . . .                                      | <a href="#">36:239</a> , <a href="#">36:240</a> , <a href="#">751</a>                                                                                                                                                                                                                                                                                                                                                                            |
| \tau . . . . .                                        | <a href="#">30:292</a>                                                                                                                                                                                                                                                                                                                                                                                                                           |

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

tbl commands:

- \tbl\_count\_missing\_cells:n . . . . .  
..... 55:134, 55:134, 55:235, 1171
- \tbl\_count\_table\_cols: . . . . .  
..... 55:128, 55:128, 1168
- \tbl\_crcr:n . . . . . 55:232, 55:232
- \tbl\_gdecr\_row\_count: 55:192, 55:198
- \tbl\_gincr\_row\_count: 55:192, 55:195
- \tbl\_gzero\_row\_count: 55:192, 55:192
- \tbl\_if\_row\_was\_started:TF . . . . .  
..... 55:186, 55:186, 55:189
- \tbl\_inbetween\_rows: . 55:201, 55:201
- \tbl\_init\_cell\_data\_for\_row: . . . . .  
..... 55:182, 55:182
- \tbl\_init\_cell\_data\_for\_table . 1165
- \tbl\_init\_cell\_data\_for\_table: . . . . .  
..... 55:158, 55:158
- \tbl\_restore\_outer\_cell\_data: . . . . .  
..... 55:204, 55:204, 1165
- \tbl\_save\_outer\_table\_cols: . . . . .  
..... 55:155, 55:155
- \tbl\_update\_cell\_data: 55:124, 55:124
- \tbl\_update\_cell\_data\_for\_next\_-  
row: . . . . . 55:178, 55:178
- \tbl\_update\_multicolumn\_cell\_-  
data:n . . . . . 55:221, 55:221

tbl internal commands:

- \g\_\_tbl\_col\_int . . . . . 55:100,  
55:125, 55:136, 55:141, 55:159,  
55:175, 55:180, 55:183, 55:187,  
55:190, 55:202, 55:205, 55:212,  
55:222, 55:225, 55:228, 55:233, 1170
- \g\_\_tbl\_missing\_cells\_int . . . . .  
..... 55:116, 55:138, 55:145, 55:149
- \g\_\_tbl\_missingcells\_int . . . . . 55:116
- \g\_\_tbl\_row\_int . . . . . 55:100,  
55:160, 55:174, 55:179, 55:193,  
55:196, 55:199, 55:206, 55:211, 1166
- \l\_\_tbl\_saved\_col\_t1 . . . . .  
..... 55:107, 55:159, 55:165, 55:205
- \l\_\_tbl\_saved\_row\_t1 . . . . .  
..... 55:107, 55:160, 55:164, 55:206
- \l\_\_tbl\_saved\_span\_t1 . . . . .  
..... 55:107, 55:161, 55:166, 55:207, 55:213
- \l\_\_tbl\_saved\_table\_cols\_t1 . . . . .  
..... 55:107, 55:156, 55:168, 55:170, 55:208
- \g\_\_tbl\_span\_t1 .. 55:100, 55:125,  
55:126, 55:142, 55:161, 55:176,  
55:184, 55:207, 55:228, 55:230, 1171
- \g\_\_tbl\_table\_cols\_t1 . . . . .  
... 55:100, 55:130, 55:131, 55:140,  
55:156, 55:208, 55:215, 55:217, 1168
- \l\_\_tbl\_tmptab\_seq 55:129, 55:130, 55:133

template internal commands:

- \\_\_template\_assign\_boolean: . . . . .  
..... 11:759, 11:759
- \\_\_template\_assign\_boolean\_aux:n  
..... 11:759, 11:762, 11:763, 11:765
- \\_\_template\_assign\_choice: . . . . .  
..... 11:785, 11:785
- \\_\_template\_assign\_choice\_-  
aux:nTF . . . . .  
11:785, 11:787, 11:790, 11:802, 11:807
- \\_\_template\_assign\_function: . . . . .  
..... 11:808, 11:808
- \\_\_template\_assign\_function\_-  
aux:N . 11:808, 11:811, 11:812, 11:814
- \\_\_template\_assign\_instance: . . . . .  
..... 11:825, 11:825
- \\_\_template\_assign\_instance\_-  
aux:N . 11:825, 11:828, 11:829, 11:831
- \\_\_template\_assign\_variable: . . . . .  
..... 11:729, 11:843, 11:843
- \\_\_template\_assign\_variable:n . . . . .  
..... 11:843, 11:845, 11:853
- \\_\_template\_assignments\_pop: . . . . .  
..... 11:908, 11:908, 11:1193
- \\_\_template\_assignments\_push:n . . . . .  
..... 11:667, 11:909, 11:909
- \l\_\_template\_assignments\_t1 . . . . .  
... 11:19, 11:668, 11:706, 11:770,  
11:778, 11:805, 11:816, 11:833,  
11:858, 11:865, 11:908, 11:910, 351
- \c\_\_template\_code\_root\_t1 . . . . .  
..... 11:9, 11:54,  
11:395, 11:398, 11:640, 11:642,  
11:643, 11:669, 11:756, 11:912, 351
- \\_\_template\_convert\_to\_assignments:  
..... 11:662, 11:704, 11:704, 11:755
- \\_\_template\_convert\_to\_assignments\_-  
aux:n . . . . . 11:704, 11:708, 11:710
- \\_\_template\_convert\_to\_assignments\_-  
aux:nn . 11:704, 11:713, 11:715, 11:734
- \\_\_template\_declare\_instance:nnnn  
..... 11:645, 11:645, 11:1169
- \\_\_template\_declare\_instance\_-  
aux:nnnn . . . . .  
..... 11:645, 11:651, 11:654, 11:701
- \\_\_template\_declare\_template\_-  
code:nnnn . . . . .  
..... 11:374, 11:384, 11:386, 11:393
- \\_\_template\_declare\_template\_-  
code:nnnnn . 11:374, 11:374, 11:1161
- \\_\_template\_declare\_template\_-  
keys:nnnn . . . 11:203, 11:203, 11:1159

**File Key:** 01=ltidirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltphyphen.dtx, 57=ltfinal.dtx

```

__template_declare_type:nn
 11:182, 11:186, 11:188
\l__template_default_tl .. 11:20, 351
\c__template_defaults_root_tl ...
 11:10,
 11:99, 11:100, 11:139, 11:142, 351
__template_define_type:nn
 11:182, 11:182, 11:1157
__template_edit_defaults:nnn ...
 11:598, 11:598, 11:1165
__template_edit_instance:nnn ...
 11:686, 11:686, 11:1173
__template_edit_instance-
 aux:mnnn ... 11:693, 11:698, 11:703
__template_edit_instance-
 aux:mnnnn 11:686
\l__template_error_bool ... 11:21,
 11:222, 11:228, 11:248, 11:277,
 11:290, 11:615, 11:656, 11:658, 351
__template_execute_if_arg-
 agree:nnTF
 11:42, 11:42, 11:207, 11:378
__template_execute_if_code-
 exist:nnTF
 11:52, 11:52, 11:647, 11:750, 11:933
__template_execute_if_keys-
 exist:nnTF 11:71
__template_execute_if_keytype-
 exist:nTF 11:58, 11:58, 11:64, 11:224
__template_execute_if_type-
 exist:nTF 11:65, 11:65, 11:205, 11:376
__template_find_global:
 11:725, 11:735, 11:735
__template_find_global_aux:w ...
 11:735, 11:740, 11:743
\l__template_global_bool
 11:22, 11:737, 11:746,
 11:761, 11:810, 11:827, 11:849, 351
__template_if_instance_exist:nn
 11:84
__template_if_instance_exist:nnTF
 ... 11:84, 11:675, 11:688, 11:904,
 11:950, 11:1187, 11:1189, 11:1191
__template_if_key_value:n
 11:77, 11:83
__template_if_key_value:nTF ...
 11:77, 11:767, 11:855
__template_if_keys_exist:nnTF ..
 . 11:71, 11:380, 11:600, 11:915, 11:924
__template_if_use_template:n .. 11:90
__template_if_use_template:nTF ..
 11:90, 11:892
__template_implement_choice-
 elt:n 11:528, 11:561, 11:593
__template_implement_choice-
 elt:nnn ... 11:529, 11:561, 11:561
__template_implement_choice-
 elt_aux:n
 11:561, 11:567, 11:575, 11:578
__template_implement_choice-
 elt_aux:nnn
 11:561, 11:566, 11:573, 11:585
__template_implement_choices:nn
 11:450, 11:522, 11:522
__template_implement_choices-
 default: ... 11:522, 11:533, 11:540
__template_instance_set_eq:nnn .
 11:673, 11:673, 11:1171
\c__template_instances_root_tl ..
 ... 11:11, 11:86, 11:663, 11:665,
 11:679, 11:681, 11:682, 11:905, 351
\l__template_key_name_tl .. 11:23,
 11:231, 11:234, 11:261, 11:263,
 11:284, 11:299, 11:306, 11:311,
 11:353, 11:355, 11:360, 11:416,
 11:419, 11:424, 11:468, 11:482,
 11:497, 11:506, 11:525, 11:526,
 11:531, 11:543, 11:547, 11:550,
 11:553, 11:556, 11:557, 11:580,
 11:583, 11:588, 11:590, 11:596,
 11:620, 11:622, 11:625, 11:727,
 11:788, 11:791, 11:793, 11:797, 352
\c__template_key_order_root_tl ..
 11:13,
 11:115, 11:116, 11:155, 11:158, 351
\l__template_key_order_seq
 ... 11:29, 11:117, 11:157, 11:160,
 11:211, 11:230, 11:263, 11:707, 352
__template_key_to_value:
 11:769, 11:857, 11:872, 11:872
__template_key_to_value_auxi:w .
 11:872, 11:873, 11:874
__template_key_to_value_auxii:w
 11:872, 11:880, 11:885
\l__template_keytype_arg_tl
 11:25, 11:245, 11:258,
 11:259, 11:266, 11:323, 11:337,
 11:463, 11:524, 11:821, 11:838, 352
\l__template_keytype_tl
 ... 11:24, 11:224, 11:243, 11:257,
 11:264, 11:273, 11:322, 11:333,
 11:420, 11:422, 11:448, 11:510,
 11:630, 11:722, 11:724, 11:728, 352
\c__template_keytypes_arg_seq ...
 11:16, 11:226, 11:346, 351
\l__template_keytypes_prop
 11:28, 11:114, 11:151,
 11:154, 11:210, 11:261, 11:409,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

11:418, 11:424, 11:550, 11:580,
11:622, 11:712, 11:793, 11:927, 352
\c__template_keytypes_root_tl ...
 ... 11:12, 11:73, 11:107, 11:110,
 11:112, 11:113, 11:149, 11:152, 351
__template_map_var_type: ...
 11:489, 11:492, 11:508, 11:508, 11:847
__template_parse_keys_elt:n ...
 .. 11:213, 11:219, 11:219, 11:272, 358
__template_parse_keys_elt:nn ...
 11:213, 11:270, 11:270
__template_parse_keys_elt_aux: ...
 11:219, 11:236, 11:253
__template_parse_keys_elt_aux:nn ...
 11:219, 11:227, 11:241
__template_parse_values:nnn ...
 11:603, 11:607, 11:607, 11:657, 11:754
__template_parse_values_elt:n ...
 11:611, 11:613, 11:613
__template_parse_values_elt:nn ...
 11:611, 11:618, 11:618
__template_parse_values_elt_-_
 aux:n 11:618, 11:624, 11:627
__template_parse_vars_elt:n ...
 11:407, 11:412, 11:412
__template_parse_vars_elt:nnn ...
 11:407, 11:414, 11:414
__template_parse_vars_elt_-_
 aux:nn 11:414, 11:423, 11:428
__template_parse_vars_elt_-_
 aux:nnn
 11:414, 11:436, 11:440, 11:446, 11:502
__template_parse_vars_elt_-_
 aux:nw 11:414, 11:430, 11:432
__template_parse_vars_elt_-_
 key:nn
 11:414, 11:455, 11:472, 11:490, 11:503
__template_quark_if_nil:N ...
__template_quark_if_nil:NTF 11:887
__template_quark_if_nil:nTF . 11:41
__template_quark_if_nil_p:n . 11:41
__template_recover_defaults:nn ...
 11:136, 11:136, 11:403,
 11:602, 11:634, 11:649, 11:752, 11:917
__template_recover_keytypes:nn ...
 11:136,
 11:146, 11:404, 11:609, 11:636, 11:926
__template_recover_values:nn ...
 11:136, 11:162, 11:677, 11:690, 11:952
__template_recover_vars:nn ...
 11:136, 11:172,
 11:638, 11:650, 11:700, 11:753, 11:935
\c__template_restrict_root_tl ... 351
__template_show:Nnnn
 11:913, 11:918, 11:927, 11:936, 11:940
__template_show_code:nn
 11:911, 11:911, 11:1177
__template_show_defaults:nn ...
 11:913, 11:913, 11:1179
__template_show_keytypes:nn ...
 11:913, 11:922, 11:1181
__template_show_values:nn ...
 11:948, 11:948, 11:1185
__template_show_vars:nn
 11:913, 11:931, 11:1183
__template_split_keytype:n ...
 11:221, 11:275, 11:275
__template_split_keytype_arg:n ...
 11:316,
 11:320, 11:320, 11:349, 11:422,
 11:552, 11:582, 11:629, 11:721, 11:795
__template_split_keytype_arg_-
 aux:n . 11:320, 11:324, 11:347, 11:350
__template_split_keytype_arg_-
 aux:w . 11:320, 11:328, 11:343, 11:351
__template_split_keytype_aux:w ...
 11:275, 11:285, 11:296, 11:308
__template_store_defaults:nn ...
 .. 11:96, 11:96, 11:214, 11:604, 11:635
__template_store_keyImplementation:nnn ...
 11:382, 11:401, 11:401
__template_store_keytypes:nn ...
 11:96, 11:104, 11:215, 11:637
__template_store_value:n
 11:354, 11:354, 11:356, 11:357, 11:358
__template_store_value_aux:Nn ...
 11:359, 11:359, 11:362,
 11:364, 11:366, 11:368, 11:370, 11:372
__template_store_value_boolean:n ...
 11:352, 11:352
__template_store_value_choice:n ...
 11:354, 11:356
__template_store_value_commaList:n ...
 11:359, 11:373
__template_store_value_function:n ...
 11:354, 11:357
__template_store_value_instance:n ...
 11:354, 11:358
__template_store_value_integer:n ...
 11:359, 11:361
__template_store_value_length:n ...
 11:359, 11:363
__template_store_value_muskip:n ...
 11:359, 11:365
__template_store_value_real:n ...
 11:359, 11:367

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

__template_store_value_skip:n 11:359, 11:369
__template_store_value_tokenlist:n 11:359, 11:371, 11:373
__template_store_values:nn 11:96, 11:120, 11:661, 11:678
__template_store_vars:nn 11:96, 11:128, 11:408, 11:639
__template_template_set_eq:nnn 11:632, 11:632, 11:1163
\l__template_tmp_clist 11:32, 11:524, 11:534, 11:536, 11:563, 11:570, 11:572, 11:1195, 352
\l__template_tmp_dim 11:33, 352
\l__template_tmp_int 11:34, 11:190, 11:191, 11:194, 11:196, 11:200, 352
\l__template_tmp_muskip 11:35, 352
\l__template_tmp_skip 11:36, 352
\l__template_tmp_tl 11:37, 11:44, 11:45, 11:49, 11:255, 11:262, 11:278, 11:279, 11:280, 11:286, 11:532, 11:542, 11:543, 11:544, 11:546, 11:547, 11:548, 11:551, 11:552, 11:554, 11:581, 11:582, 11:589, 11:591, 11:623, 11:629, 11:692, 11:694, 11:712, 11:713, 11:794, 11:795, 11:804, 11:805, 11:876, 11:877, 11:883, 352
\g__template_type_prop 11:18, 11:44, 11:67, 11:184, 11:195, 351
__template_use_instance:nn 11:837, 11:890, 11:890, 11:1175
__template_use_instance_aux:nn 11:890, 11:894, 11:902
__template_use_instance_-aux:nNnn 11:890, 11:893, 11:896
__template_use_template:nnn 11:748, 11:748, 11:899, 11:1167
\l__template_value_tl 11:26, 11:717, 11:767, 11:774, 11:780, 11:788, 11:798, 11:822, 11:839, 11:855, 11:861, 11:868, 11:873, 11:878, 11:880, 11:888, 352
\l__template_values_prop 11:30, 11:101, 11:125, 11:141, 11:144, 11:167, 11:170, 11:209, 11:353, 11:355, 11:360, 11:531, 11:553, 11:660, 11:691, 11:717, 11:918, 11:957, 361
\c__template_values_root_tl 11:14, 11:123, 11:124, 11:165, 11:168, 351
\l__template_var_tl 11:27, 11:719, 11:738, 11:740, 11:745, 11:773, 11:781, 11:819, 11:835, 11:860, 11:867, 352
\l__template_vars_prop 11:31, 11:133, 11:177, 11:180, 11:405, 11:467, 11:481, 11:496, 11:525, 11:544, 11:548, 11:591, 11:719, 11:804, 11:877, 11:936, 352
\c__template_vars_root_tl 11:15, 11:131, 11:132, 11:175, 11:178, 351
\tencirc 31:10, 42:125, 42:677
\tencircw 31:10, 42:128
\tenln 31:9, 42:124, 42:126, 42:676, 42:678
\tenlnw 31:9, 42:127, 42:129
\test 1204
\testallgroups 33:1250, 33:1423, 33:1424, 33:1425, 33:1427, 33:1428, 33:1429, 33:1430, 33:1431, 33:1432, 33:1433, 33:1434, 33:1435, 33:1436, 33:1437, 33:1438, 33:1439, 33:1440, 33:1441, 33:1442, 33:1443, 33:1444, 33:1445, 33:1449, 33:1450, 33:1451, 33:1452, 33:1453, 33:1454, 33:1458, 33:1459, 33:1460, 33:1464, 33:1465, 33:1466, 33:1467, 33:1468, 33:1469, 33:1470, 33:1471, 33:1472, 33:1473, 33:1474, 33:1475, 33:1476, 33:1477, 33:1478, 33:1479, 33:1480, 33:1481, 33:1482, 33:1483, 33:1484, 33:1486, 33:1490, 33:1491, 33:1492, 33:1493, 33:1494, 33:1495, 33:1496, 33:1497, 33:1498, 33:1500, 33:1501, 33:1502, 33:1503, 33:1504, 33:1505, 33:1506, 33:1507, 33:1508, 33:1509, 33:1510, 33:1511, 33:1512, 33:1513, 33:1514, 33:1515, 33:1516, 33:1517, 33:1518, 33:1519, 33:1520, 33:1521, 33:1522, 33:1523, 33:1524, 33:1525, 33:1526, 33:1527, 33:1528, 33:1529, 33:1530, 33:1531, 33:1532, 33:1533, 33:1534, 33:1535, 33:1536, 33:1537, 33:1538, 33:1539, 33:1540, 33:1541, 33:1542, 33:1543, 33:1544, 33:1545, 33:1546, 33:1547, 33:1548, 33:1549, 33:1550, 33:1551, 33:1552, 33:1553, 33:1554, 33:1555, 33:1556, 33:1557, 33:1558, 33:1559, 33:1560, 33:1561, 33:1562, 33:1563, 33:1564, 33:1565, 33:1566, 33:1567, 33:1568, 33:1569, 33:1570, 33:1571, 33:1572, 33:1573, 33:1574, 33:1575, 33:1576, 33:1577, 33:1578, 33:1579, 33:1580, 33:1581, 33:1582, 33:1583, 33:1584, 33:1585, 33:1586, 33:1587, 33:1588, 33:1589, 33:1590, 33:1591, 33:1592

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

33:1593, 33:1594, 33:1595, 33:1596,  
 33:1597, 33:1598, 33:1599, 33:1600,  
 33:1601, 33:1602, 33:1603, 33:1604,  
 33:1605, 33:1606, 33:1607, 33:1608,  
 33:1609, 33:1610, 33:1611, 33:1612,  
 33:1613, 33:1614, 33:1615, 33:1634  
`\testallkerneldefinedfamilies` . . . . .  
                                                   33:1422, 33:1631  
`\testFont` 33:1218, 33:1219, 33:1251, 33:1252  
`\testgroup` . 33:1223, 33:1261, 33:1270,  
                                                   33:1276, 33:1282, 33:1286, 33:1290,  
                                                   33:1300, 33:1304, 33:1372, 33:1376  
`\TeX` . . . . . 19:1, 19:12, 1249  
`\TeX` and `\LaTeX` 2 $\varepsilon$  commands:  
   `\...-h@k` . . . . . 1249  
   `\...@without@substitution` . . . . . 577  
   `\@` . . . . . 09:375,  
     14:19, 04:20, 18:500, 19:2, 04:1026,  
     50:43, 50:58, 50:71, 50:80, 50:118,  
     06:833, 06:834, 57:512, 01:49, 1199  
   `\@...hook` . . . . . 229  
   `\@?@?` . . . . . 1199  
   `\@@` . 13:15, 13:19, 13:20, 13:21, 13:22,  
     13:24, 13:27, 13:28, 13:30, 13:31,  
     20:753, 20:773, 26:510, 26:512,  
     26:513, 41:238, 41:239, 41:240,  
     41:250, 01:315, 01:316, 54:10, 54:11  
   `\@@DeclareMathDelimiter` . . . . . 1210  
   `\@@defaultsubs` . . . . . 24:610  
   `\@@enc@update` . . . . .  
     . . . . . 21:183, 24:259, 24:263, 1222  
   `\@@end` . . . . . 20:722, 20:723,  
     01:206, 37:33, 37:98, 37:174, 37:240,  
     06:23, 56:18, 57:667, 57:688, 01:53, 462  
   `\@@endpbox` . . . . . 41:193, 41:236, 41:386  
   `\@@eqncr` . . . . .  
     38:381, 38:403, 38:413, 38:418, 38:541  
   `\@@fileswith@pti@ns` . . . . .  
     . . . . . 50:605, 50:624, 50:1104  
   `\@@hyph` . . . . . 06:24, 06:850  
   `\@@hyphenation` . . . . . 21:205  
   `\@@if@newlist` . . . . .  
     . . . . . 54:603, 54:648, 54:661, 54:707  
   `\@@ifdefinable` . . . . . 21:17, 06:132  
   `\@@input` . . . . . 20:620,  
     20:633, 20:673, 37:27, 37:92,  
     37:154, 06:22, 50:1727, 52:172,  
     52:188, 52:196, 57:331, 01:52, 1204  
   `\@@italiccorr` 32:113, 32:117, 06:25, 1206  
   `\@@line` . . . . . 40:614  
   `\@@math@bgroup` . . . . . 32:131, 32:138  
   `\@@math@egroup` . . . . . 32:128  
   `\@@par` . 15:4, 16:140, 16:177, 37:240,  
     37:531, 37:536, 37:539, 37:553, 37:557, 37:560, 39:82, 39:85, 06:21,  
     40:334, 40:361, 40:385, 40:406,  
     41:199, 44:67, 44:118, 54:259, 1232  
`\@@patterns` . . . . . 21:205  
`\@@protect` . . . . . 06:266, 06:272, 06:281  
`\@@sqrt` . . . . . 1231  
`\@@startpbox` . . . . . 41:193, 41:236, 41:386  
`\@@sverb` . . . . . 37:646, 781  
`\@@text@case@aux` . . . . .  
     . . . . . 57:577, 57:608, 57:613, 57:618  
`\@@underline` . . . . . 40:551, 40:554, 40:555  
`\@@unprocessedoptions` . . . . .  
     . . . . . 50:1016, 50:1081, 50:1179  
`\@@warning` . . . . . 14:215, 1202  
`\@Alph` . . . . . 22:188, 22:204, 109  
`\@DeclareEncodingSubset` . . . . . 33:45,  
     33:47, 33:48, 33:49, 33:50, 33:53, 33:60  
`\@DeclareMathDelimiter` . . . . .  
     . . . . . 28:967, 28:986, 1210  
`\@DeclareMathSizes` . . . . .  
     . . . . . 24:206, 24:207, 24:209  
`\@Ephack` . . . . .  
     . . . . . 18:191, 45:201, 45:223, 45:241, 1196  
`\@Ialph` . . . . . 1212  
`\@IncludeInRelease@se` . . . . . 03:71  
`\@IncludeInRelease` . . . . . 03:71  
`\@LRmoderr` 14:271, 18:239, 18:290, 1269  
`\@M` . . . . . 18:11, 18:12, 18:13, 18:14,  
     18:15, 18:16, 18:17, 18:18, 18:19,  
     18:120, 24:679, 24:687, 26:439,  
     26:452, 38:391, 39:218, 06:37, 06:39,  
     41:56, 44:67, 44:100, 44:118, 44:130,  
     44:209, 44:232, 54:178, 54:197,  
     54:200, 54:260, 02:21, 02:508, 02:509  
`\@MM` . . . . . 45:469,  
     45:488, 45:506, 54:301, 02:21, 382  
`\@Mi` . . . . . 12:3, 54:138  
`\@Mii` . . . . .  
     12:3, 45:53, 45:122, 45:194, 45:216,  
     45:241, 45:311, 54:297, 54:1163, 54:1330  
`\@Miii` 12:3, 45:55, 45:124, 45:313, 54:300  
`\@Miv` . . . . . 12:3,  
     45:195, 45:201, 45:217, 45:223, 54:274  
`\@Roman` . . . . . 22:186, 22:192, 1235  
`\@TeXversion` . . . . . 14:28, 01:310, 2  
`\@abs@page@last` . . . . . 53:105, 53:111,  
     53:351, 53:353, 53:355, 53:363,  
     53:367, 53:370, 53:409, 53:410, 1080  
`\@acci` . . . . . 29:775, 40:387, 40:408, 1207  
`\@accii` . . . . . 29:775, 40:387, 40:408, 1212  
`\@acciii` . . . . . 29:775, 40:387, 40:408, 1212  
`\@acol` . . . . . 41:168, 41:178,  
     41:260, 41:261, 41:273, 41:274,  
     41:277, 41:294, 41:309, 41:317, 41:327

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphyphen.dtx, 57=ltfinal.dtx

\@acolampacol .... 41:258, 41:275,  
   41:277, 41:284, 41:292, 41:326, 41:329  
 \@activechar@info ..... 54:579, 1209  
 \@activechar@warning ..... 1209  
 \@addamp ..... 41:251, 41:260,  
   41:261, 41:276, 41:290, 41:327, 41:328  
 \@addfield ..... 41:43, 41:53,  
   41:86, 41:93, 41:125, 41:140, 41:142  
 \@addmarginpar ..... 54:333, 54:1815  
 \@addtobot ..... 54:979, 54:1066,  
   54:1133, 54:1185, 54:1294, 54:1353  
 \@addtocurcol 54:330, 54:1070, 54:2003  
 \@addtoblcol ..... 54:858, 54:1566  
 \@addtofilelist .....  
   ... 20:64, 20:133, 20:188, 20:620,  
   20:742, 29:749, 29:752, 29:759,  
   29:762, 29:769, 29:772, 50:1726,  
   52:169, 52:188, 52:196, 57:279,  
   57:282, 57:708, 01:85, 01:87, 1206  
 \@addtonextcol 54:857, 54:1390, 54:2004  
 \@addtopreamble .. 41:311, 41:324,  
   41:330, 41:331, 41:332, 41:334, 41:346  
 \@addtoreset ..... 22:16, 22:44,  
   22:64, 22:77, 22:137, 22:163, 22:166  
 \@addtotoporbot ..... 54:1016,  
   54:1179, 54:1347, 54:1439, 54:1528  
 \@afterheading ... 44:92, 44:125, 1233  
 \@afterindentfalse ..... 44:45  
 \@afterindenttrue .....  
   ... 44:43, 44:124, 44:208, 44:231  
 \@alph .... 22:187, 22:200, 45:399, 109  
 \@ampacol 41:258, 41:275, 41:286, 41:329  
 \@arabic ..... 22:52, 22:70,  
   22:159, 22:171, 22:184, 22:190, 45:397  
 \@argarraycr ..... 41:203, 41:204  
 \@argdef ..... 06:80  
 \@argsrbox ..... 40:575  
 \@argtabularcr ..... 41:210, 41:211  
 \@array ..... 41:181, 41:182  
 \@arrayacol ..... 41:168, 41:258  
 \@arrayclassiv ..... 41:169, 41:331  
 \@arrayclassz ..... 41:168, 41:275  
 \@arraycr ..... 41:170, 41:201, 41:203  
 \@arrayparboxrestore .....  
   ... 40:377, 40:419, 41:384  
 \@arrayrule ..... 41:309,  
   41:311, 41:315, 41:317, 41:319, 41:346  
 \@arstrut ..... 41:192, 41:237, 41:343  
 \@arstrutbox .....  
   . 41:185, 41:218, 41:343, 41:385, 1210  
 \@author ..... 44:8, 44:32  
 \@auxout ... 20:217, 20:223, 20:267,  
   20:285, 20:309, 20:342, 20:371,  
   20:394, 20:418, 20:433, 35:105,  
   35:165, 36:55, 44:181, 47:7, 47:8,  
   47:39, 47:49, 47:57, 47:67, 47:84, 53:362  
 \@backslashchar .. 14:234, 14:236,  
   30:273, 06:231, 50:1308, 50:1441,  
   06:481, 50:1530, 06:630, 06:642,  
   06:646, 06:647, 06:652, 06:696, 1212  
 \@badcrerr ..... 14:279, 1210  
 \@badend ..... 14:247, 37:390, 1197  
 \@badlinearg ..... 14:267, 42:165,  
   42:177, 42:188, 42:189, 42:193,  
   42:242, 42:247, 42:257, 42:262, 42:275  
 \@badmath ..... 14:251, 38:275,  
   38:277, 38:282, 38:285, 38:294,  
   38:306, 38:311, 38:320, 38:333,  
   38:338, 38:464, 38:476, 38:492, 38:501  
 \@badpoptabs ..... 14:255, 41:85, 41:151  
 \@badrequireerror .... 50:497, 50:1187  
 \@badtab ..... 14:258, 41:22,  
   41:87, 41:108, 41:114, 41:121, 41:148  
 \begin@tempboxa .....  
   40:27, 40:46, 40:61, 40:246, 40:265,  
   40:334, 40:361, 40:576, 40:584, 1201  
 \begindocumenthook .....  
   ... 20:61, 20:127, 20:130, 20:182,  
   20:185, 47:53, 50:1115, 50:1136, 216  
 \begindvi 54:627, 54:686, 54:714, 1066  
 \begindvibox .....  
   ... 53:449, 53:450, 54:86, 54:715, 1061  
 \begin{parpenalty} .. 18:15, 38:467,  
   38:479, 38:505, 39:23, 39:194, 803  
 \begin{theorem} ..... 43:43, 43:48  
 \bezier ..... 42:683, 42:682  
 \bibitem ..... 47:3, 47:8  
 \biblabel ..... 47:4, 47:97  
 \bitor ..... 54:15, 54:885,  
   54:905, 54:941, 54:964, 54:1031,  
   54:1115, 54:1125, 54:1273, 54:1284,  
   54:1426, 54:1513, 54:1631, 54:1756  
 \botlist .....  
   ... 54:65, 54:386, 54:388, 54:433,  
   54:435, 54:721, 54:742, 54:751,  
   54:752, 54:993, 54:996, 54:1031,  
   54:1125, 54:1284, 54:1959, 54:1987  
 \botnum ..... 45:274,  
   54:111, 54:990, 54:991, 54:996,  
   54:1000, 54:1462, 54:1467, 54:1555,  
   54:1562, 54:1951, 54:1979, 54:2021  
 \botroom ..... 45:275,  
   54:112, 54:993, 54:996, 54:1952, 54:1980  
 \boxfpsbit . 54:2069, 54:2071, 54:2076  
 \break@loop ..... 1210  
 \break@tfor .....  
   ... 13:31, 20:593, 20:610, 32:98, 1210

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

\@bsphack . . . . . 18:37,  
   18:126, 18:387, 18:403, 18:421,  
   18:437, 35:102, 35:164, 36:82, 45:52,  
   45:121, 45:310, 46:6, 46:18, 46:23,  
   46:35, 47:63, 47:80, 54:1886, 74:9  
\@caption . . . . . 45:12, 45:14  
\@captype . . . . . 45:5, 45:9, 45:12,  
   45:40, 45:88, 45:109, 45:157, 54:2033  
\@car . . . . . 19:14, 21:89, 21:110, 06:53  
\@carcube . . . . . 06:55, 06:135, 06:633  
\@cclv . . . . . 54:302,  
   54:306, 54:384, 54:385, 54:414,  
   54:431, 54:432, 54:461, 54:487,  
   54:491, 54:492, 02:16, 57:67, 1066  
\@cclvi 04:30, 04:58, 50:1306, 50:1439,  
   50:1528, 02:21, 02:57, 02:82,  
   02:93, 02:95, 02:99, 02:159, 02:173  
\@cdr . . . . . 20:251, 06:53, 06:776, 06:777  
\@centercr . . . . . 37:413, 37:451, 37:456,  
   37:461, 37:471, 37:475, 37:479, 1250  
\@centering . . . . . 38:360, 38:361,  
   38:369, 38:372, 38:375, 38:534, 38:538  
\@cf1b . . . . . 54:718  
\@cf1t . . . . . 54:718  
\@changed@cmd . . . . . 21:3, 21:63,  
   21:223, 24:131, 24:267, 57:476, 1223  
\@changed@x . . . . . 21:3, 21:211, 21:219, 1222  
\@changed@x@err . . . . . 1230  
\@changed@x@mouth . . . . . 21:211, 21:219, 1222  
\@charlb . . . . . 20:439, 20:447  
\@charrb . . . . . 20:441, 20:447  
\@chclass . . . . .  
   41:271, 41:272, 41:335, 41:348, 41:353  
\@check@IncludeInRelease . . . . . 03:71  
\@check@c . . . . . 06:189, 06:191  
\@check@eq . . . . . 06:195, 06:196, 06:200  
\@checkcommand . . . . . 1214  
\@checkend . . . . .  
   . . . . . 37:16, 37:82, 37:146, 37:337,  
   37:352, 37:369, 37:379, 37:389, 1224  
\@chnum . . . . . 41:279,  
   41:298, 41:335, 41:350, 41:351, 41:352  
\@circ . . . . .  
   42:621, 42:639, 42:648, 42:653, 42:656  
\@circle . . . . . 42:604, 42:605  
\@circlefnt . . . . . 42:125,  
   42:128, 42:447, 42:490, 42:519,  
   42:544, 42:614, 42:630, 42:662, 42:677  
\@cite . . . . . 47:36, 47:95  
\@cite@ofmt . . . . . 47:44, 47:96  
\@citea . . . . . 47:35, 47:37  
\@citeb . . . . . 47:38, 47:39,  
   47:40, 47:43, 47:44, 47:66, 47:67,  
   47:68, 47:69, 47:83, 47:84, 47:85, 47:86  
\@citex . . . . . 47:20, 47:21, 47:29, 47:34, 938  
\@citex@checkblank . . . . . 47:17, 47:18, 47:30  
\@classi . . . . . 41:271, 41:307  
\@classii . . . . . 41:271, 41:321  
\@classiii . . . . . 41:271, 41:326  
\@classiv . . . . . 41:169, 41:180, 41:272  
\@classoptionslist . . . . .  
   . . . . . 50:9, 50:546, 50:561, 50:562,  
   50:579, 50:810, 50:811, 50:839,  
   50:840, 50:866, 50:867, 50:1808, 1257  
\@classv . . . . . 41:272, 41:332  
\@classz . . . . . 41:168, 41:179, 41:271  
\@cline . . . . . 41:367  
\@clnht 42:195, 42:196, 42:204, 42:206,  
   42:208, 42:218, 42:225, 42:273, 42:671  
\@clnwd . . . . . 42:197,  
   42:203, 42:207, 42:209, 42:210, 42:671  
\@cls@pkg . . . . . 08:2673,  
   50:353, 50:354, 50:367, 50:368,  
   50:955, 50:965, 50:1013, 50:1060,  
   50:1090, 50:1142, 50:1172, 50:1174,  
   50:1191, 50:1736, 50:1758, 50:1788  
\@clsextension . . . . . 50:31,  
   50:156, 50:164, 50:222, 50:383,  
   50:399, 50:411, 50:493, 50:517,  
   50:528, 50:546, 50:560, 50:578,  
   50:677, 50:692, 50:716, 50:809,  
   50:838, 50:865, 50:959, 50:1006,  
   50:1033, 50:1094, 50:1107, 50:1147,  
   50:1162, 50:1613, 51:99, 51:165, 1046  
\@clubpenalty . . . . . 20:7, 20:25, 20:95,  
   20:152, 39:128, 39:220, 44:106, 44:135  
\@colht . . . . . 20:22, 20:92, 20:149,  
   45:273, 45:275, 45:278, 45:284,  
   45:285, 45:298, 45:299, 54:116,  
   54:233, 54:244, 54:253, 54:254,  
   54:389, 54:401, 54:436, 54:449,  
   54:478, 54:509, 54:539, 54:545,  
   54:549, 54:559, 54:564, 54:649,  
   54:708, 54:781, 54:819, 54:863,  
   54:888, 54:907, 54:947, 54:969,  
   54:1646, 54:1772, 54:2134, 57:155  
\@colnum . . . . .  
   . . . . . 45:276, 54:113, 54:999, 54:1044,  
   54:1113, 54:1114, 54:1142, 54:1150,  
   54:1271, 54:1272, 54:1304, 54:1316,  
   54:1424, 54:1425, 54:1462, 54:1467,  
   54:1511, 54:1512, 54:1554, 54:1561,  
   54:1947, 54:1975, 54:2014, 54:2189  
\@colroom . . . . . 20:23, 20:93,  
   20:150, 54:117, 54:254, 54:275,  
   54:276, 54:287, 54:290, 54:389,  
   54:436, 54:781, 54:998, 54:1043,  
   54:1109, 54:1112, 54:1141, 54:1266,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltyphephen.dtx, 57=ltfinal.dtx

54:1270, 54:1303, 54:1420, 54:1423,  
 54:1506, 54:1510, 54:1948, 54:1976,  
 54:2144, 54:2149, 54:2194, 57:154, 1231  
`\@combinedblfloats` .....  
     ..... 54:754, 54:2268, 54:2307  
`\@combinefloats` ..... 54:505, 54:718  
`\@comdblflelt` ..... 54:754  
`\@comflelt` ..... 54:724, 54:740, 54:754  
`\@compatibility` ..... 1204  
`\@cons` ..... 22:81,  
     22:90, 06:52, 45:193, 45:215, 45:239,  
     45:379, 54:239, 54:892, 54:911,  
     54:927, 54:951, 54:953, 54:973,  
     54:975, 54:1145, 54:1213, 54:1309,  
     54:1382, 54:1455, 54:1545, 54:1648,  
     54:1671, 54:1774, 54:1799, 54:1816,  
     54:1817, 54:2195, 02:196, 02:214  
`\@contentsline@destination` .....  
     ..... 44:188, 44:190, 44:197, 908  
`\@contfield` ..... 41:50, 41:141, 41:153  
`\@copy@...` ..... 100  
`\@copy@DeclareRobustCommand` .....  
     06:497, 06:567, 06:588, 06:666, 06:669  
`\@copy@newcommand` .....  
     ..... 06:318, 06:498, 06:567,  
     06:609, 06:639, 06:666, 06:672, 93  
`\@copytexsys` ..... 1204  
`\@ctrerr` .....  
     14:243, 22:203, 22:207, 22:221, 22:229  
`\@curfield` ..... 41:16, 41:41, 41:47,  
     41:51, 41:52, 41:54, 41:130, 41:131  
`\@curline` ..... 41:16, 41:27,  
     41:39, 41:44, 41:53, 41:54, 41:55,  
     41:90, 41:91, 41:103, 41:128, 41:129  
`\@curr@enc` ..... 21:154, 21:156  
`\@curr@file` .....  
     ... 20:227, 20:228, 20:237, 20:239,  
     20:263, 20:271, 20:452, 20:471,  
     20:640, 20:655, 50:940, 50:1253,  
     50:1258, 50:1264, 50:1270, 50:1274,  
     50:1285, 50:1294, 50:1321, 50:1384,  
     50:1389, 50:1395, 50:1418, 50:1427,  
     50:1453, 52:266, 52:374, 52:376, 1049  
`\@curr@file@reqd` .....  
     ..... 52:266, 52:376, 52:380, 1049  
`\@currbox` ..... 45:60,  
     45:91, 45:95, 45:129, 45:160, 45:164,  
     45:193, 45:214, 45:215, 45:239,  
     45:257, 45:259, 45:261, 45:319,  
     45:322, 45:327, 45:331, 54:215,  
     54:216, 54:227, 54:228, 54:230,  
     54:231, 54:239, 54:313, 54:314,  
     54:857, 54:858, 54:1106, 54:1108,  
     54:1116, 54:1139, 54:1143, 54:1145,  
     54:1160, 54:1201, 54:1213, 54:1261,  
     54:1264, 54:1301, 54:1306, 54:1309,  
     54:1326, 54:1371, 54:1382, 54:1414,  
     54:1430, 54:1444, 54:1455, 54:1497,  
     54:1534, 54:1545, 54:1585, 54:1589,  
     54:1600, 54:1606, 54:1608, 54:1612,  
     54:1617, 54:1626, 54:1635, 54:1641,  
     54:1648, 54:1671, 54:1706, 54:1710,  
     54:1722, 54:1729, 54:1731, 54:1735,  
     54:1741, 54:1751, 54:1766, 54:1774,  
     54:1799, 54:1817, 54:1826, 54:2039,  
     54:2040, 54:2069, 54:2099, 54:2104,  
     54:2150, 54:2153, 54:2165, 54:2173,  
     54:2190, 54:2195, 02:275, 02:276, 02:277  
`\@currdir` .....  
     ... 01:92, 01:114, 01:116, 01:122,  
     01:124, 01:130, 01:132, 01:137,  
     01:139, 01:149, 01:162, 01:227,  
     01:240, 01:253, 50:1232, 50:1258,  
     50:1367, 50:1389, 50:1418, 50:1501, 9  
`\@current@cmd` ..... 21:25, 24:271, 1222  
`\@currentHpage` ..... 36:240, 36:241, 751  
`\@currentHref` ..... 17:34,  
     17:38, 35:99, 35:107, 35:116, 35:120,  
     35:140, 36:239, 55:53, 55:57, 751  
`\@currentcounter` .. 35:113, 35:125,  
     35:135, 35:142, 35:171, 35:173,  
     35:183, 36:242, 38:365, 38:519,  
     40:485, 43:53, 43:55, 45:471, 1260  
`\@currentlabel` ... 35:106, 35:126,  
     35:136, 35:143, 35:166, 35:174,  
     35:185, 35:196, 35:204, 36:236,  
     38:364, 38:518, 40:486, 40:503,  
     40:520, 45:472, 45:490, 45:508, 795  
`\@currentlabelname` .....  
     ..... 35:98, 35:107, 36:238, 751  
`\@currenttarget` ..... 35:162  
`\@currenttitle` ..... 35:161  
`\@currenvir` .....  
     ... 14:249, 37:3, 37:251, 37:272,  
     37:290, 37:300, 37:390, 39:112,  
     40:188, 50:1296, 50:1308, 50:1316,  
     50:1320, 50:1327, 50:1429, 50:1441,  
     50:1449, 50:1453, 50:1459, 50:1518,  
     50:1530, 50:1538, 50:1542, 50:1548, 407  
`\@currenvline` ..... 14:249, 37:252,  
     37:273, 37:291, 37:301, 37:391, 40:189  
`\@currext` ..... 50:30, 50:42, 50:57,  
     50:70, 50:79, 50:117, 50:379, 50:382,  
     50:383, 50:398, 50:399, 50:410,  
     50:411, 50:517, 50:528, 50:539,  
     50:546, 50:560, 50:578, 50:607,  
     50:687, 50:700, 50:702, 50:710,  
     50:724, 50:908, 50:909, 50:911,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

50:915, 50:918, 50:920, 50:925,  
 50:930, 50:932, 50:937, 50:943,  
 50:951, 50:957, 50:959, 50:963,  
 50:968, 50:974, 50:983, 50:986,  
 50:991, 50:994, 50:997, 50:999,  
 50:1000, 50:1002, 50:1006, 50:1015,  
 50:1017, 50:1018, 50:1023, 50:1026,  
 50:1029, 50:1033, 50:1039, 50:1052,  
 50:1057, 50:1058, 50:1063, 50:1069,  
 50:1073, 50:1075, 50:1076, 50:1078,  
 50:1080, 50:1082, 50:1083, 50:1086,  
 50:1092, 50:1094, 50:1107, 50:1120,  
 50:1147, 50:1150, 50:1162, 50:1180,  
 50:1181, 51:62, 51:99, 51:106,  
 51:108, 51:165, 51:167, 51:170, 1002  
 \@currextension ..... 1198  
 \@currlist ..... 45:193,  
 45:215, 45:379, 54:67, 54:313,  
 54:390, 54:393, 54:437, 54:440, 54:1816  
 \@currname .....  
 ... 03:71, 03:116, 03:124, 20:762,  
 20:784, 20:790, 50:29, 50:41, 50:56,  
 50:69, 50:78, 50:116, 50:351, 50:353,  
 50:365, 50:367, 50:379, 50:382,  
 50:398, 50:410, 50:539, 50:607,  
 50:700, 50:702, 50:710, 50:724,  
 50:906, 50:909, 50:911, 50:915,  
 50:918, 50:920, 50:925, 50:927,  
 50:930, 50:932, 50:937, 50:942,  
 50:951, 50:955, 50:957, 50:963,  
 50:965, 50:966, 50:968, 50:974,  
 50:983, 50:985, 50:991, 50:994,  
 50:997, 50:999, 50:1000, 50:1004,  
 50:1008, 50:1015, 50:1017, 50:1018,  
 50:1023, 50:1026, 50:1030, 50:1034,  
 50:1039, 50:1051, 50:1075, 50:1076,  
 50:1078, 50:1080, 50:1082, 50:1083,  
 50:1120, 50:1172, 50:1174, 50:1181,  
 50:1191, 50:1736, 50:1758, 50:1788,  
 51:62, 51:75, 51:106, 51:108, 51:167,  
 51:170, 51:184, 51:186, 51:197,  
 51:255, 08:368, 08:374, 08:429, 1002  
 \@currnamestack .....  
 ... 50:33, 50:136, 52:546, 08:426, 982  
 \@curroption ..... 1198  
 \@curroptions ..... 50:539,  
 50:547, 50:597, 50:616, 50:1181, 50:1182  
 \@currpath .....  
 ... 50:15, 50:46, 50:139, 50:351,  
 50:353, 50:907, 50:925, 50:932,  
 50:941, 50:983, 50:984, 50:1011, 980  
 \@currpkgrqnd ..... 50:381,  
 50:935, 50:937, 50:946, 50:979,  
 50:993, 50:996, 50:1011, 50:1013, 1003  
 \@currsize ..... 29:639, 1197  
 \@currtype ..... 54:121, 54:882,  
 54:883, 54:884, 54:885, 54:902,  
 54:903, 54:904, 54:905, 54:1031,  
 54:1115, 54:1125, 54:1273, 54:1284,  
 54:1426, 54:1513, 54:1631, 54:1756,  
 54:2039, 54:2041, 54:2042, 54:2045  
 \@curtab 41:11, 41:26, 41:86, 41:87,  
 41:88, 41:94, 41:95, 41:98, 41:102,  
 41:103, 41:107, 41:146, 41:147, 1229  
 \@curtabmar 41:11, 41:25, 41:26, 41:38,  
 41:44, 41:89, 41:102, 41:106, 41:107  
 \@dcr ..... 01:145, 01:146  
 \@dashbox .....  
 ... 42:324, 42:325, 42:326, 42:327,  
 42:328, 42:331, 42:337, 42:339,  
 42:349, 42:351, 42:352, 42:353,  
 42:354, 42:358, 42:362, 42:365,  
 42:382, 42:384, 42:385, 42:386,  
 42:387, 42:390, 42:394, 42:396,  
 42:405, 42:407, 42:408, 42:409,  
 42:410, 42:413, 42:417, 42:422, 42:673  
 \@dashcnt .....  
 ... 42:317, 42:319, 42:320, 42:321,  
 42:322, 42:323, 42:336, 42:338,  
 42:341, 42:343, 42:344, 42:345,  
 42:347, 42:348, 42:361, 42:364,  
 42:376, 42:377, 42:378, 42:379,  
 42:380, 42:381, 42:393, 42:395,  
 42:398, 42:399, 42:400, 42:401,  
 42:403, 42:404, 42:416, 42:421, 42:673  
 \@dashdim ..... 42:316, 42:317,  
 42:318, 42:319, 42:320, 42:322,  
 42:325, 42:327, 42:328, 42:329,  
 42:336, 42:338, 42:340, 42:341,  
 42:342, 42:343, 42:344, 42:347,  
 42:351, 42:353, 42:354, 42:355,  
 42:363, 42:366, 42:375, 42:376,  
 42:377, 42:378, 42:380, 42:384,  
 42:386, 42:387, 42:388, 42:393,  
 42:395, 42:397, 42:398, 42:399,  
 42:400, 42:403, 42:407, 42:409,  
 42:410, 42:411, 42:419, 42:424, 42:673  
 \@date ..... 44:9, 44:33  
 \@dbflt ..... 45:32, 45:264, 1218  
 \@dblarg .. 44:54, 44:142, 45:12, 06:798  
 \@dbldeferlist ..... 45:239,  
 54:70, 54:447, 54:452, 54:454,  
 54:820, 54:827, 54:828, 54:1756,  
 54:1799, 54:1963, 54:1992, 1243  
 \@dblfloating ..... 45:31  
 \@dblfloatingplacement 20:31, 20:101,  
 20:159, 45:280, 54:403, 54:451,  
 54:1944, 54:1972, 54:2273, 54:2313

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@dblflset 45:26
\@dblfpbot 45:290, 45:304, 54:2357
\@dblfpsep 45:289, 45:303, 54:2357
\@dblfpstop 45:288, 45:302, 54:2357
\@dbltoplist 54:69, 54:234,
 54:237, 54:239, 54:399, 54:400,
 54:447, 54:448, 54:759, 54:763,
 54:765, 54:766, 54:1643, 54:1648,
 54:1768, 54:1774, 54:1962, 54:1990
\@dbltopnum 45:283,
 45:297, 54:109, 54:129, 54:240,
 54:242, 54:770, 54:1582, 54:1583,
 54:1647, 54:1650, 54:1678, 54:1683,
 54:1703, 54:1704, 54:1773, 54:1777,
 54:1806, 54:1811, 54:1955, 54:1983
\@dbltoproom 45:284, 45:286, 45:298,
 45:300, 54:110, 54:1585, 54:1588,
 54:1589, 54:1598, 54:1599, 54:1602,
 54:1605, 54:1608, 54:1612, 54:1616,
 54:1620, 54:1625, 54:1645, 54:1706,
 54:1709, 54:1710, 54:1719, 54:1720,
 54:1721, 54:1724, 54:1728, 54:1731,
 54:1735, 54:1740, 54:1744, 54:1749,
 54:1750, 54:1771, 54:1956, 54:1984
\@dec@text@cmd 21:3
\@declarecommandcopylisthook ...
 .. 06:494, 06:496, 06:508, 07:1127, 99
\@declaredoptions 50:8, 50:500, 50:543,
 50:581, 50:602, 50:621, 50:1113, 1198
\@declareoption 50:498, 50:499, 50:507
\@defaultfamilyhook 29:429, 29:664,
 29:681, 29:696, 29:701, 29:716, 1251
\@defaultsubs 24:564, 24:598,
 24:610, 37:57, 37:122, 37:161, 1227
\@defaultunits 24:214, 24:218, 24:219, 24:220,
 24:235, 24:328, 26:179, 26:181, 42:13
\@defaultunitsset ... 40:74, 40:85,
 42:8, 42:29, 42:30, 42:32, 42:34,
 42:60, 42:63, 42:84, 42:85, 42:107,
 42:108, 42:164, 42:241, 42:316,
 42:318, 42:332, 42:340, 42:342,
 42:357, 42:479, 42:480, 42:611,
 42:647, 42:689, 42:690, 42:692,
 42:693, 42:696, 42:697, 42:699,
 42:700, 42:711, 42:712, 42:714,
 42:715, 42:717, 42:718, 42:720, 42:721
\@defdefault@ds 50:498, 50:503, 50:508
\@deferlist 54:68,
 54:386, 54:395, 54:396, 54:399,
 54:404, 54:406, 54:412, 54:433,
 54:442, 54:444, 54:782, 54:790,
 54:791, 54:802, 54:807, 54:808,
 54:1115, 54:1213, 54:1273, 54:1382,
 54:1426, 54:1455, 54:1513, 54:1545,
 54:1631, 54:1671, 54:1961, 54:1989
\@definecounter 22:12, 22:36,
 38:349, 39:251, 39:252, 39:253,
 39:254, 43:8, 43:16, 45:396, 45:398
\@depth 26:191, 30:565, 30:566, 30:568,
 30:569, 06:26, 40:550, 40:603,
 40:610, 41:187, 41:219, 42:227,
 42:300, 42:303, 42:324, 42:333,
 42:383, 42:391, 42:726, 42:782, 54:1855
\@dir 01:144, 01:147, 01:149, 01:151, 01:152
\@disable@packageload@do 50:930, 52:477
\@dischypth 40:386, 40:407, 06:835, 06:869, 1208
\@docclearpage 54:298, 54:373
\@documentclass 1200
\@documentclass@hook 50:3, 50:815, 50:843, 50:870, 1205
\@doendpe 37:338, 37:353,
 37:370, 37:380, 39:123, 39:152, 811
\@ofilelist 20:759,
 20:801, 37:42, 37:107, 37:156, 1222
\@ofilelist@hash 20:790, 20:796
\@ofilelist@size 20:784, 20:796
\@donoparitem 39:168, 39:182
\@dot 42:604, 42:642
\@dotsep 44:215, 44:238
\@dottedtocline 44:200, 44:226, 44:227
\@downline 42:297, 42:301, 42:306
\@downvector 42:268, 42:306
\@eha 03:195, 14:219, 14:237, 14:239,
 14:241, 14:250, 14:252, 14:284,
 20:224, 20:268, 20:286, 21:52,
 21:84, 24:28, 24:58, 24:102, 24:144,
 24:187, 24:253, 24:339, 26:106,
 28:25, 28:70, 28:99, 28:172, 28:203,
 28:233, 28:265, 28:478, 28:499,
 28:551, 28:602, 28:647, 28:652,
 28:707, 28:825, 28:829, 28:833,
 28:868, 28:872, 28:876, 28:933,
 28:943, 28:1028, 28:1033, 28:1036,
 28:1068, 28:1071, 28:1144, 28:1147,
 28:1150, 28:1217, 28:1223, 32:146,
 33:24, 33:81, 33:912, 33:921, 37:250,
 37:271, 37:289, 37:299, 47:71, 47:88,
 06:292, 06:331, 06:359, 54:1880, 54:1896
\@ehb 14:219, 14:244, 14:270, 14:272,
 14:274, 14:276, 54:236, 54:392, 54:439
\@ehc 03:184, 14:219,
 14:279, 14:282, 14:288, 14:290,
 37:655, 37:672, 37:686, 37:701,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

37:714, 38:422, 39:244, 06:128,  
 06:155, 44:31, 06:481, 50:1788, 06:522  
 \@ehd ..... 03:102, 03:159,  
 14:219, 14:246, 14:254, 14:257,  
 14:259, 14:265, 28:118, 05:88,  
 41:100, 41:109, 45:6, 50:757, 50:1013  
 \@elt ..... 20:440, 21:1576,  
 21:1578, 22:20, 22:35, 22:101,  
 22:104, 29:141, 29:152, 29:154,  
 29:155, 29:180, 29:505, 29:507,  
 29:508, 29:518, 29:742, 30:17, 30:25,  
 06:52, 54:8, 54:11, 54:15, 54:27,  
 54:30, 54:31, 54:32, 54:33, 54:38,  
 54:39, 54:40, 54:41, 54:42, 54:43,  
 54:44, 54:45, 54:47, 54:51, 54:57,  
 54:58, 54:59, 54:60, 54:502, 54:724,  
 54:735, 54:740, 54:750, 54:762,  
 54:764, 54:792, 54:809, 54:829,  
 54:848, 54:861, 54:868, 54:919,  
 54:922, 54:931, 54:1922, 54:1934, 651  
 \@empty ..... 13:14, 1197  
 \@emptycol ..... 54:200,  
 54:247, 54:250, 54:279, 54:283, 1209  
 \@end@check@IncludeInRelease ...  
 ..... 03:143, 03:145  
 \@end@tempboxa .....  
 ..... 40:36, 40:55, 40:64, 40:258,  
 40:270, 40:352, 40:374, 40:582, 40:592  
 \@enddocument@kernel@warnings ...  
 .. 37:43, 37:52, 37:108, 37:117, 37:176  
 \@enddocumenthook .....  
 ..... 37:145, 50:1115, 50:1137  
 \@endfloatbox .....  
 ..... 45:190, 45:211, 45:236, 45:248  
 \@endparenv ..... 39:120, 39:123, 811  
 \@endparpenalty ..... 18:16, 38:468,  
 38:480, 38:506, 39:23, 39:124, 803  
 \@endpbox ..... 41:193,  
 41:236, 41:266, 41:333, 41:384, 41:387  
 \@endpefalse 37:256, 37:276, 37:294,  
 37:304, 39:129, 39:133, 39:137,  
 39:138, 39:140, 40:138, 40:191, 1201  
 \@endpetrue .....  
 .. 39:124, 39:126, 39:136, 39:140, 813  
 \@endpreamblehook ..... 446  
 \@endtheorem 43:13, 43:19, 43:25, 43:48  
 \@enlargepage 54:1865, 54:1870, 54:1872  
 \@ensuredmath ..... 38:433, 38:435  
 \@enumctr ..... 39:258, 39:261, 39:262  
 \@enumdepth .....  
 .. 39:250, 39:256, 39:257, 39:258, 816  
 \@enumspacing ..... 816  
 \@eqcnt ..... 38:357, 38:419,  
 38:424, 38:521, 38:536, 38:537, 38:539  
 \@eqnccr .....  
 .. 38:370, 38:388, 38:425, 38:426, 38:523  
 \@eqnnum ..... 38:351,  
 38:352, 38:423, 38:454, 38:513, 1196  
 \@eqnsel ..... 38:357, 38:535, 1198  
 \@eqnswfalse ..... 38:387  
 \@eqnswtrue .....  
 .. 38:359, 38:366, 38:424, 38:520  
 \@eqopen .....  
 .. 38:357, 38:391, 38:393, 38:404, 38:414  
 \@er@ext ..... 1199  
 \@err@ ..... 14:37, 14:41,  
 14:44, 14:52, 14:64, 14:68, 14:71, 14:79  
 \@esphack ... 18:39, 18:132, 18:392,  
 18:409, 18:426, 18:443, 35:109,  
 35:167, 36:86, 45:385, 46:17, 46:19,  
 46:34, 47:74, 47:90, 54:1888, 1196  
 \@evenfoot . 49:12, 49:15, 54:617, 54:676  
 \@evenhead . 49:12, 49:15, 54:616, 54:675  
 \@execute@begin@hook .....  
 .. 37:253, 37:257, 37:260, 769  
 \@executeoption ..... 1199  
 \@expandtwoargs .....  
 .. 03:188, 06:217, 50:224,  
 50:350, 50:545, 50:581, 50:635, 50:644  
 \@expast ..... 41:239, 41:267  
 \@expl@@filehook@clear@replacement@flag@@  
 ..... 50:449,  
 50:472, 52:295, 52:318, 52:339, 52:519  
 \@expl@@filehook@drop@extension@on  
 ..... 52:292, 52:293, 52:315,  
 52:316, 52:336, 52:337, 52:521, 1050  
 \@expl@@filehook@file@pop@@ ...  
 .. 50:948, 52:157, 52:525, 52:535  
 \@expl@@filehook@file@pop@assign@onnnn  
 ..... 52:162, 52:527, 1046  
 \@expl@@filehook@file@push@@ ...  
 .. 50:931, 52:151, 52:523  
 \@expl@@filehook@if@file@replaced@onTF  
 .. 52:289, 52:312, 52:333, 52:517, 1049  
 \@expl@@filehook@if@no@extension@onTF  
 ..... 52:285,  
 52:308, 52:329, 52:507, 52:509, 52:534  
 \@expl@@filehook@normalize@file@name@onw  
 .. 52:291, 52:314, 52:335, 52:515  
 \@expl@@filehook@resolve@file@subst@onw  
 .. 50:447,  
 50:470, 52:288, 52:311, 52:332, 52:513  
 \@expl@@filehook@set@curr@file@onNN  
 .. 50:446, 50:469,  
 50:900, 52:374, 52:380, 52:511, 1049  
 \@expl@@hook@curr@name@pop@@ ...  
 .. 08:2880, 50:89  
 \@expl@@hook@curr@name@push@on 1255

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@expl@@@initialize@call@@ \@failedlist

..... 08:2880, 37:261, 08:1469 ... 54:846, 54:869, 54:885, 54:892,

\@expl@@@mark@update@dblcol@structures@@ 54:905, 54:911, 54:927, 54:941, 54:964

..... 48:477, 48:490, 54:470 \@fcollmadefalse 54:837

\@expl@@@mark@update@singlocol@structures@@@fcollmadetru 54:925

..... 48:475, 48:489, 54:473 \@file-subst@{file} 1047

\@expl@@@shipout@add@background@box@@n \@filef@und . 20:491, 20:503, 20:504,

..... 53:417, 53:502 20:507, 20:513, 20:545, 20:567,

\@expl@@@shipout@add@background@picture@@n 20:590, 20:607, 20:620, 20:673,

..... 53:417, 53:506 52:155, 52:172, 52:188, 52:196, 1045

\@expl@@@shipout@add@firstpage@material@@N\@filehook@file@push 1002

..... 53:417, 53:499 \@filehook@set@CurrentFile

\@expl@@@shipout@add@foreground@box@@n 20:315, 20:376, 50:933, 52:152, 52:363

..... 53:417, 53:504 \@filelist .. 20:63, 20:132, 20:187,

\@expl@@@shipout@add@foreground@picture@@n 20:741, 20:742, 20:761, 29:749,

..... 53:417, 53:508 29:759, 29:769, 57:279, 57:692, 57:708

\@expl@char@generate@nn \@filesfalse 20:200,

..... 05:145, 06:814, 1254 50:1228, 50:1229, 50:1363, 50:1364

\@expl@cs@(thing)@spec@@N 1253 \@fileswith@pti@ns 50:497,

\@expl@cs@argument@spec@@N 1268 50:605, 50:624, 50:801, 50:802,

\@expl@cs@parameter@spec@@N 50:806, 50:808, 50:836, 50:837,

... 05:140, 05:141, 05:142, 05:153, 50:863, 50:864, 50:891, 50:1104, 1198

06:657, 06:710, 06:721, 06:726, 1268 \@fileswith@ptions 50:786,

\@expl@cs@prefix@spec@@N 06:725 50:787, 50:794, 50:795, 50:799, 50:803

..... 05:138, 05:152, 06:655, 06:725 \@fileswithoptions

\@expl@cs@replacement@spec@@N 50:677, 50:684, 50:692, 50:775

..... 05:143, 05:154, 06:615, \@filestrue 20:5,

06:650, 06:657, 06:711, 06:722, 06:726 50:1211, 50:1214, 50:1271, 50:1275,

\@expl@cs@to@str@@N 06:1346, 50:1349, 50:1404, 50:1408

..... 05:133, 05:136, 05:148, 05:150, \@finalstrut

06:489, 06:581, 06:601, 06:603, ... 40:490, 40:507, 40:524, 40:593,

06:604, 06:617, 06:630, 06:642, 41:385, 45:477, 45:495, 45:513, 1211

06:646, 06:647, 06:652, 06:696, \@firststampfalse 41:254, 41:277, 41:294

06:705, 06:709, 06:717, 06:719, 1253 \@firststamptrue 41:262

\@expl@finalise@setup@@ \@firstcolfirstmark

..... 05:30, 57:268, 57:269 54:2250, 54:2251, 54:2255

\@expl@pop@filename@@ \@firstcoltopmark 54:2248, 54:2256

..... 05:26, 50:88, 50:92, 50:98, 50:109, 1253 \@firstcolumnfalse .. 54:2240, 54:2285

\@expl@push@filename@@ \@firstcolumntrue .. 20:28, 20:98,

..... 05:24, 50:37, 50:39, 50:52, 20:156, 54:100, 54:209, 54:2259, 54:2291

50:54, 50:64, 50:76, 50:96, 50:102, 1253 \@firstoffive 35:23, 35:25, 35:40, 35:57

\@expl@push@filename@aux@@ \@firstofone 04:12, 20:126, 20:181,

..... 08:2778, 05:25, 50:37, 04:100, 04:108, 20:459, 21:68,

50:48, 50:60, 50:64, 50:82, 50:102, 298 21:153, 04:166, 26:346, 28:53, 28:81,

\@expl@str@if@eq@@nnTF ... 05:137, 28:146, 28:184, 28:214, 28:245,

..... 05:151, 50:350, 06:453, 06:455, 1253 28:978, 05:34, 05:61, 05:76, 37:144,

\@expl@str@map@function@@NN 38:431, 41:372, 45:10, 06:212, 47:38,

..... 05:144, 05:155, 06:811, 109 47:66, 47:83, 50:949, 50:978, 06:477,

\@expl@str@map@function@@NN and 06:482, 06:483, 06:486, 06:487,

\@expl@char@generate@nn ... 1254 06:518, 06:523, 06:524, 06:527,

\@expl@sys@load@backend@@ 06:528, 52:352, 52:369, 57:359, 1224

..... 20:17, 05:21, 05:23 \@firstoftwo

\@expl@text@uppercase@@n 03:85, 17:17, 20:506, 20:512,

\@extra@page@added 37:68, 37:133, 53:387 20:546, 20:591, 20:608, 21:133,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspace.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

22:239, 22:244, 28:982, 29:510,  
 33:69, 33:834, 33:884, 33:900, 35:42,  
 05:35, 35:78, 05:43, 06:212, 49:16,  
 50:161, 50:195, 50:207, 50:230,  
 50:248, 50:307, 50:337, 06:454,  
 06:456, 06:478, 50:1797, 06:519,  
 06:584, 06:635, 06:750, 06:760,  
 06:770, 06:797, 06:821, 01:71, 460  
 \@firsttab ..... 41:2, 41:74,  
 41:75, 41:76, 41:106, 41:118, 1210  
 \@flcheckspace 54:993, 54:1029, 54:2140  
 \@flfail ..... 54:869,  
 54:920, 54:941, 54:951, 54:964, 54:973  
 \@float ..... 45:26, 45:32  
 \@floatboxreset 45:101, 45:170, 45:174  
 \@floatpenalty .....  
 . 45:3, 45:53, 45:55, 45:58, 45:122,  
 45:124, 45:127, 45:191, 45:194,  
 45:199, 45:201, 45:212, 45:216,  
 45:221, 45:223, 45:237, 45:241,  
 45:311, 45:313, 45:317, 45:321, 45:379  
 \@floatplacement ... 20:31, 20:101,  
 20:159, 45:271, 54:151, 54:211,  
 54:255, 54:481, 54:1945, 54:1973, 1200  
 \@flsetnum ..... 54:990,  
 54:1026, 54:1113, 54:1271, 54:1424,  
 54:1511, 54:1582, 54:1703, 54:2108  
 \@flsettextmin ..... 54:1089,  
 54:1241, 54:1410, 54:1493, 54:2124  
 \@flstop ..... 54:2010  
 \@flsucceed .....  
 54:862, 54:870, 54:919, 54:953, 54:975  
 \@fltovf . 14:275, 45:93, 45:162, 45:322  
 \@flupdates .. 54:996, 54:1041, 54:2186  
 \@flushglue . 12:17, 37:452, 37:456,  
 37:462, 37:472, 37:475, 37:480,  
 37:530, 37:552, 39:76, 40:393, 40:414  
 \@fnssymbol ..... 22:189, 22:208  
 \@font@aliasinfo ..... 26:539  
 \@font@info .....  
 . 24:133, 24:171, 24:177, 24:391,  
 24:408, 24:646, 25:478, 26:30,  
 26:38, 26:46, 26:74, 26:87, 26:154,  
 26:200, 26:214, 26:225, 26:239,  
 26:255, 26:261, 26:274, 26:281,  
 26:288, 26:293, 26:303, 26:315,  
 26:327, 26:491, 26:503, 26:508,  
 26:515, 26:545, 26:558, 26:566,  
 28:279, 28:290, 28:299, 28:312,  
 28:329, 28:338, 28:353, 28:368,  
 28:426, 28:482, 28:581, 28:587,  
 28:631, 28:644, 28:727, 28:816,  
 28:859, 28:924, 28:1018, 28:1185,  
 28:1214, 33:55, 33:56, 33:99, 57:478

\@font@series@contextfalse .....  
 ..... 29:502, 29:538  
 \@font@series@contexttrue .....  
 ..... 29:521, 29:525, 29:537  
 \@font@shape@subst@warning .....  
 ..... 25:445, 25:448,  
 25:453, 25:511, 25:658, 25:661, 569  
 \@font@warning ..... 24:3,  
 24:560, 24:565, 24:592, 24:599,  
 25:456, 26:19, 26:33, 26:41, 26:49,  
 26:61, 26:77, 26:476, 26:490, 26:502,  
 26:507, 26:514, 26:557, 26:565,  
 27:30, 37:54, 37:119, 37:158, 57:300  
 \@fontenc@load@list .....  
 . 21:1577, 29:742, 30:17, 30:25, 1251  
 \@fontswitch ..... 32:126, 32:128, 1210  
 \@footnotemark ..... 45:454,  
 45:460, 45:522, 45:528, 45:529, 45:555  
 \@footnotetext ..... 40:455,  
 45:454, 45:460, 45:461, 45:538, 45:544  
 \@for ..... 13:16, 20:233,  
 20:305, 20:367, 20:414, 20:761,  
 47:36, 47:65, 47:82, 50:232, 50:249,  
 50:543, 50:562, 50:579, 50:597,  
 50:602, 50:616, 50:621, 50:657,  
 50:667, 50:1182, 50:1219, 50:1354, 938  
 \@forced@seriesfalse .....  
 ..... 25:394, 25:407, 26:140  
 \@forced@seriestrue .. 25:398, 25:409  
 \@forloop ..... 13:19, 13:20  
 \@fornoop ..... 13:15, 13:23, 13:29  
 \@fortmp .....  
 13:17, 13:18, 13:26, 50:655, 50:657,  
 50:1218, 50:1219, 50:1353, 50:1354  
 \@fpbot 45:290, 45:304, 54:867, 54:2351  
 \@fpmin ..... 45:278,  
 45:287, 45:301, 54:115, 54:924,  
 54:1953, 54:1981, 54:2203, 54:2220  
 \@fps 45:41, 45:42, 45:44, 45:47, 45:64,  
 45:110, 45:111, 45:113, 45:116,  
 45:133, 54:2031, 54:2033, 54:2036  
 \@fpsaddrdefault .....  
 . 45:45, 45:48, 45:114, 45:117, 54:2028  
 \@fpsep ..... 45:289, 45:303,  
 54:865, 54:874, 54:946, 54:968, 54:2351  
 \@fpstype ..... 54:987,  
 54:1008, 54:1009, 54:1023, 54:1054,  
 54:1055, 54:1079, 54:1081, 54:1084,  
 54:1086, 54:1137, 54:1193, 54:1194,  
 54:1229, 54:1232, 54:1235, 54:1238,  
 54:1299, 54:1361, 54:1362, 54:1400,  
 54:1402, 54:1405, 54:1407, 54:1481,  
 54:1484, 54:1487, 54:1490, 54:1579,  
 54:1594, 54:1596, 54:1614, 54:1623,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

54:1659, 54:1660, 54:1700, 54:1715,  
 54:1717, 54:1737, 54:1747, 54:1786,  
 54:1787, 54:2024, 54:2040, 54:2042,  
 54:2044, 54:2047, 54:2048, 54:2049,  
 54:2051, 54:2052, 54:2056, 54:2057,  
 54:2059, 54:2060, 54:2094, 54:2096,  
 54:2098, 54:2110, 54:2112, 54:2126,  
 54:2128, 54:2158, 54:2161, 54:2172  
`\@fptop` 45:288, 45:302, 54:864, [54:2351](#)  
`\@frameb@x` . . . . . 40:218, 40:257, 40:269, [40:273](#), [1217](#)  
`\@framebox` . . . . . 40:225, 40:232, [40:236](#)  
`\@framepicbox` . . . . . 40:225, 40:232, [40:294](#)  
`\@freelist` . . . . . 45:60,  
 45:129, 45:319, 45:320, 54:29, 54:34,  
 54:48, 54:56, 54:215, 54:503, 54:736,  
 54:751, 54:765, 54:870, 54:1816,  
 54:1817, 02:196, 02:214, 02:275, [1244](#)  
`\@generic@error` . . . . . [1215](#)  
`\@generic@message` . . . . . [1215](#)  
`\@getcirc` . . . . . [42:437](#),  
 42:484, 42:513, 42:540, 42:612, 42:628  
`\@getfpsbit` . . . . . 54:984,  
 54:1020, 54:1576, 54:1697, 54:2067  
`\@getlarrow` . . . . . 42:266, 42:274, [42:276](#)  
`\@getlinechar` . . . . . 42:190, [42:229](#)  
`\@getpen` . . . . . 18:35, 18:38, 18:47, [18:118](#)  
`\@getrarrow` . . . . . 42:267, 42:274, [42:283](#)  
`\@glossaryfile` . . . . . 46:21, 46:22, 46:31  
`\@gnewline` 18:96, 18:102, 18:109, [18:112](#)  
`\@gobble` 03:163, 13:6, 13:9, 03:187,  
 14:101, 14:127, 14:147, 14:155,  
 14:180, 14:189, 14:202, 17:15,  
 18:76, 04:11, 18:559, 20:64, 20:133,  
 20:188, 04:98, 20:466, 20:468,  
 20:741, 21:29, 24:561, 24:594,  
 26:345, 27:26, 28:28, 28:30, 28:430,  
 28:441, 28:525, 28:592, 28:593,  
 28:622, 28:628, 28:636, 28:641,  
 28:659, 28:673, 28:683, 28:692,  
 28:705, 28:722, 28:731, 28:805,  
 28:807, 28:811, 28:819, 28:853,  
 28:862, 28:914, 28:916, 28:927,  
 28:1011, 28:1021, 28:1102, 28:1107,  
 28:1176, 28:1207, 29:139, 29:180,  
 29:518, 29:752, 29:762, 29:772,  
 33:861, 05:33, 05:65, 05:88, 37:281,  
 06:111, 06:133, 44:143, 44:144,  
 44:145, 44:146, 44:147, 44:182,  
[06:208](#), 45:7, 06:231, 06:248, 06:252,  
 47:11, 47:45, 47:46, 06:289, 06:295,  
 06:298, 06:308, 06:328, 06:334,  
 06:337, 06:346, 06:356, 06:362,  
 06:365, 06:374, 06:392, 06:396,  
 06:398, 06:399, 06:401, 06:409,  
 06:413, 06:415, 50:745, 50:928,  
 50:1203, 50:1267, 50:1294, 50:1398,  
 50:1427, 50:1511, 50:1516, 50:1590,  
 50:1742, 50:1754, 52:261, 54:623,  
 54:624, 54:625, 54:682, 54:683,  
 54:684, 54:931, 54:1936, 54:2204,  
 54:2221, 57:282, 57:411, 57:708, [1211](#)  
`\@gobble@AddToHook@args` . . . . .  
 . . . . . 08:2896, 08:2897  
`\@gobble@IncludeInRelease` . . . . . [03:71](#)  
`\@gobble@RemoveFromHook@arg` . . . . .  
 . . . . . 08:2899, 08:2900  
`\@gobblecr` . . . . . 18:557, 18:558  
`\@gobblefour` . . . . . 28:24, 28:427,  
 28:583, 28:585, 28:589, 28:591,  
 28:601, 28:605, 28:729, 28:781,  
[06:208](#), 50:1296, 50:1429, 50:1518  
`\@gobblethree` . . . . .  
 . . . . . [06:208](#), 50:128, 06:644, 06:656, [1222](#)  
`\@gobbletwo` . . . . . 13:12,  
 20:32, 20:102, 20:160, 24:566,  
 24:600, 28:132, 37:23, 37:55, 37:89,  
 37:120, 37:151, 37:159, 05:123,  
 05:128, 06:175, 06:176, 06:208,  
 49:11, 49:13, 50:1266, 50:1397,  
 50:1510, 52:260, 52:489, 57:306, [1227](#)  
`\@gtempa` . . . . . 20:685,  
 20:686, 20:692, 20:693, 20:694,  
 20:719, 20:720, 20:722, 20:723,  
 20:724, 41:3, 41:5, 41:6, 41:7,  
 41:8, 06:126, 06:127, 06:181, 06:183,  
 50:349, 50:351, 50:364, 50:365,  
 50:384, 50:386, 50:400, 50:402,  
 50:412, 50:414, 06:575, 06:583, [1206](#)  
`\@halfwidth` . . . . . [42:2](#),  
 42:126, 42:129, 42:131, 42:227,  
 42:299, 42:302, 42:324, 42:333,  
 42:349, 42:361, 42:364, 42:383,  
 42:391, 42:405, 42:416, 42:421,  
 42:679, 42:705, 42:724, 42:725,  
 42:726, 42:765, 42:780, 42:781, 42:782  
`\@halignto` 41:170, 41:174, 41:177, 41:191  
`\@hangfrom` . . . . . 44:66, 44:117, [44:138](#)  
`\@height` . . . . .  
 . . . . . 18:397, 18:405, 18:431, 18:439,  
 21:293, 21:295, 26:190, 30:347,  
 30:565, 30:566, 30:568, 30:569,  
[06:26](#), 40:202, 40:207, 40:280,  
 40:290, 40:550, 40:603, 40:610,  
 41:186, 41:219, 41:359, 41:376,  
 42:227, 42:300, 42:303, 42:324,  
 42:333, 42:351, 42:359, 42:383,  
 42:391, 42:407, 42:414, 42:589,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

42:599, 42:725, 42:781, 54:1855, 02:502  
 \@highpenalty ..... 18:119, 57:3  
 \@hightab ..... 41:11,  
     41:21, 41:23, 41:74, 41:86, 41:95,  
     41:96, 41:111, 41:146, 41:147, 1229  
 \@hline . 42:167, 42:179, 42:226, 42:265  
 \@holdpg ..... 54:124, 54:302,  
     54:304, 54:305, 54:310, 54:311, 54:312  
 \@hspace . 18:512, 18:513, 18:529, 1197  
 \@hspacer ..... 18:512, 18:528, 1197  
 \@hvector ..... 42:244, 42:259, 42:265  
 \@ialph ..... 1212  
 \@icentercr ..... 37:430, 37:431  
 \@iden ..... 06:215  
 \@if ..... 06:171, 06:172, 06:174  
 \if... ..... 1212  
 \if@DeclareRobustCommand .....  
     ..... 09:126, 06:497,  
     06:554, 06:565, 06:566, 06:570,  
     06:664, 06:665, 06:668, 06:691, 101  
 \if@newcommand .. 09:127, 09:138,  
     06:316, 06:498, 06:555, 06:566,  
     06:608, 06:620, 06:626, 06:692, 103  
 \if@newlist ..... 1239  
 \if@pti@ns .....  
     50:224, 50:227, 50:229, 50:246, 50:247  
 \if@ptions .....  
     50:221, 50:222, 50:223, 50:963, 50:1058  
 \if@short@command ..... 1212  
 \ifatmargin ..... 41:55, 41:106  
 \ifbothcounters .....  
     ..... 22:109, 22:120, 22:128,  
     22:136, 22:152, 22:154, 22:163, 22:165  
 \ifclasslater . 50:163, 50:171, 50:179  
 \ifclassloaded 50:155, 50:268, 50:277  
 \ifclasswith .. 50:221, 50:270, 50:279  
 \ifdefinable .....  
     ..... 21:14, 21:17, 22:11, 23:3,  
     29:618, 40:154, 06:84, 06:86, 06:130,  
     06:132, 43:7, 43:15, 43:22, 06:250, 1212  
 \iffileonpath .....  
     20:487, 20:541, 20:563, 20:580, 20:601  
 \ifl@aded ..... 50:155, 50:156,  
     50:157, 50:909, 50:1039, 50:1057, 1002  
 \ifl@t@r .....  
     .. 21:1538, 05:201, 05:207, 50:169,  
     50:172, 50:177, 50:180, 50:184,  
     50:188, 50:190, 50:201, 50:202, 50:766  
 \ifl@ter ..... 21:1583, 21:1584,  
     50:163, 50:164, 50:183, 50:951, 50:1086  
 \ifl@ter@0 .. 21:1583, 21:1584  
 \ifnch .....  
     .. 05:49, 05:50, 06:782, 06:784, 06:796

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@ignoretrue 18:201, 18:214, 24:358, 37:4, 37:7,
38:348, 38:351, 38:384, 38:544, 1232
\@iiiminipage 40:427, 40:431, 40:434, 40:435, 40:436
\@iiiparbox 40:305, 40:313, 40:320, 40:323, 40:324, 40:325, 40:472
\@iiminipage 40:430, 40:432
\@iinput 20:633, 20:634, 460
\@iiparbox 40:319, 40:321
\@iirsbox 40:574, 40:583
\@imakebox 40:26, 40:41, 40:177
\@imakepicbox 40:68, 40:69, 40:182, 40:297
\@iminipage 40:426, 40:428
\@in@minipage@envtrue 40:451
\@include 20:228, 20:271, 20:287, 20:291, 1252
\@includeinreleasefalse 03:74, 03:79, 03:133, 03:141, 06:873
\@includeinreleasetrue 03:123
\@index 46:18, 46:19, 46:35
\@indexfile 46:4, 46:5, 46:14
\@inlabel 802
\@inlabelfalse 39:28, 39:104, 39:208, 54:165, 54:192, 1207
\@inlabeltrue 39:28, 39:202
\@inmatherr 14:285, 39:112, 39:166, 42:604, 1209
\@inmathwarn 21:3
\@inpenc@test 57:356, 57:423
\@input 20:34, 20:104, 20:162, 20:299, 20:361, 20:408, 20:672, 44:152, 57:711, 1036
\@input@ 20:319, 20:346, 20:379, 20:398, 20:423, 20:674, 24:418, 47:51, 1203
\@input@file@exists@with@hooks 52:143, 1255
\@inputcheck 20:482, 20:483, 20:490, 20:536, 20:537, 20:544, 20:558, 20:559, 20:566, 20:588, 20:589, 20:592, 20:605, 20:606, 20:609, 01:175, 01:176, 01:179, 01:187, 05:38, 05:39, 05:42, 06:38, 06:45, 50:1255, 50:1256, 50:1290, 50:1386, 50:1387, 50:1423, 50:1498, 50:1499, 50:1506, 01:54, 02:307, 1257
\@insertfalse ... 54:1077, 54:1227, 54:1398, 54:1479, 54:1574, 54:1695
\@inserttrue ... 54:1003, 54:1048, 54:1165, 54:1333, 54:1653, 54:1780
\@invalidchar 14:290
\@iparbox 40:304, 40:312, 40:317
\@iirsbox 40:562, 40:569, 40:574, 40:575
\@isavebox 40:175, 40:176
\@isavepicbox 40:180, 40:181
\@ishortstack 42:133, 42:141
\@istackcr 42:143, 42:144
\@itabcr 41:57, 41:58
\@item 39:167, 39:180
\@itemdepth 39:265, 39:267, 39:268, 39:269, 816
\@itemfudge 41:38, 41:44, 41:82
\@itemitem 39:269, 39:272
\@itemlabel 39:44, 39:96, 39:167, 1201
\@itempenalty 18:17, 39:23, 39:199, 803
\@itemspacing 816
\@iwhiledim 13:7
\@iwhilenum 13:3
\@iwhilesw 13:10
\@ixpt 24:743
\@ixstackcr 42:142
\@kernel@... 980
\@kernel@Ref 35:151, 35:156
\@kernel@after@<hook> 215
\@kernel@after@begindocument 09:61, 20:58, 20:76, 05:7, 51:235, 1257
\@kernel@after@begindocument@before 20:16, 20:76, 25:731, 446
\@kernel@after@enddocument 05:1, 37:15, 37:81, 53:352
\@kernel@after@enddocument@afterlastpage 05:1, 37:19, 37:85, 53:358, 08:455
\@kernel@after@para@after 16:8, 16:106, 405
\@kernel@after@para@end 16:8, 16:99, 405
\@kernel@after@shipout@background 53:72, 53:160, 1256
\@kernel@after@shipout@lastpage 53:114, 53:119, 53:160, 53:378, 53:383, 1081
\@kernel@before@<hook> 215
\@kernel@before@begindocument 20:56, 20:76, 05:7, 53:407, 1257
\@kernel@before@enddocument 37:13, 37:79, 37:180, 765
\@kernel@before@insertmark 48:221, 48:249, 959
\@kernel@before@para@before 16:8, 16:21, 16:54, 404
\@kernel@before@para@begin 16:8, 16:30, 16:62, 405
\@kernel@before@shipout@background 53:68, 53:70, 53:160, 1069
\@kernel@currentdata 35:163

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@kernel@currpathstack
 50:45, 50:47, 50:91, 50:123, 982
\@kernel@eqno . . . 38:443, 38:445, 38:451
\@kernel@leqno . . . 38:444, 38:446, 38:452
\@kernel@make@file@csname
 52:287, 52:290,
 52:310, 52:313, 52:331, 52:334, 52:363
\@kernel@new@label@record@testdef
 36:149, 36:279, 37:25
\@kernel@pageref 35:65, 35:69
\@kernel@pageref@exp 35:70
\@kernel@ref 35:64, 35:67, 35:151
\@kernel@ref@exp 35:72
\@kernel@refstepcounter
 35:132, 35:146, 43:31, 1267
\@kernel@rename@newcommand . . .
 06:297, 06:314,
 06:351, 06:379, 06:384, 06:397, 95
\@kernel@reserved@label@data . . .
 35:100, 35:107
\@kernel@sRef 35:153, 35:156
\@kernel@spageref
 35:26, 35:43, 35:60, 35:65, 35:69
\@kernel@sref 35:13, 35:25, 35:30,
 35:42, 35:47, 35:59, 35:64, 35:67, 35:153
\@killglue
 42:59, 42:73, 42:103, 42:115, 42:123
\@kludgeins 54:321, 54:322, 54:323,
 54:325, 54:378, 54:379, 54:425,
 54:426, 54:506, 54:522, 54:523,
 54:529, 54:530, 54:531, 54:540,
 54:556, 54:560, 54:570, 54:1856, 54:1887
\@labels 39:27, 39:170,
 39:171, 39:213, 39:230, 39:231, 802
\@largefloatcheck
 45:192, 45:213, 45:238, 45:256, 1210
\@lastchclass 41:262, 41:272,
 41:273, 41:275, 41:283, 41:308,
 41:322, 41:326, 41:335, 41:348, 41:349
\@latex@error
 08:2923, 03:100, 03:159, 03:184,
 03:195, 14:163, 14:217, 14:233,
 14:239, 14:241, 14:244, 14:246,
 14:248, 14:252, 14:254, 14:256,
 14:259, 14:263, 14:268, 14:272,
 14:274, 14:276, 14:278, 14:279,
 14:281, 14:284, 14:288, 14:290,
 17:10, 20:224, 20:268, 20:286,
 21:50, 21:84, 24:5, 24:28, 24:58,
 24:102, 24:144, 24:187, 24:253,
 24:339, 26:105, 27:100, 27:111,
 28:23, 28:68, 28:97, 28:117, 28:170,
 28:201, 28:231, 28:263, 28:364,
 28:380, 28:478, 28:499, 28:551,
 28:601, 28:605, 28:647, 28:652,
 28:707, 28:775, 28:781, 28:825,
 28:829, 28:833, 28:868, 28:872,
 28:876, 28:933, 28:943, 28:1028,
 28:1033, 28:1036, 28:1068, 28:1071,
 28:1144, 28:1147, 28:1150, 28:1217,
 28:1223, 29:49, 29:60, 29:82, 29:93,
 29:600, 29:720, 32:143, 05:80,
 37:250, 37:271, 37:289, 37:299,
 37:655, 37:672, 37:686, 37:701,
 37:713, 38:422, 39:243, 41:100,
 41:109, 06:128, 06:155, 44:31,
 45:6, 45:83, 47:71, 47:88, 06:290,
 06:329, 06:357, 50:688, 50:739,
 50:752, 50:781, 50:964, 50:1012,
 50:1059, 50:1171, 50:1188, 50:1196,
 50:1201, 50:1223, 50:1280, 50:1358,
 50:1413, 06:481, 50:1757, 50:1787,
 06:522, 54:236, 54:392, 54:1878, 54:1895
\@latex@info 14:163,
 14:208, 21:85, 33:76, 06:236, 06:310,
 06:348, 06:376, 52:487, 06:835, 394
\@latex@info@no@line
 14:163, 14:209, 53:89, 54:580, 1214
\@latex@note 14:190, 24:33
\@latex@note@no@line
 14:190, 50:1237, 50:1257, 50:1263, 1258
\@latex@warning
 14:163, 14:215, 20:344,
 20:396, 21:55, 22:54, 22:72, 35:18,
 35:35, 35:52, 36:211, 36:221, 36:227,
 40:51, 40:252, 40:347, 40:447,
 42:449, 45:260, 47:42, 47:69, 47:86,
 50:1319, 50:1326, 50:1452, 50:1458,
 50:1541, 50:1547, 53:174, 54:2034
\@latex@warning@no@line . . .
 14:163, 14:216, 20:19,
 20:89, 20:146, 20:739, 35:8, 35:88,
 35:89, 36:133, 36:134, 05:98, 37:62,
 37:69, 37:127, 37:134, 37:166, 44:32,
 06:202, 50:352, 50:366, 50:767,
 50:952, 50:1087, 50:1239, 50:1388,
 50:1394, 50:1417, 50:1500, 50:1507,
 50:1574, 50:1658, 53:79, 53:98,
 53:369, 54:245, 54:277, 54:1831, 54:2100
\@latexbug 14:277, 54:335, 54:1817
\@latexerr 14:215, 54:439, 1211
\@latexinfo 1209
\@latexrelease@catcode@null . . .
 07:6, 07:3015
\@lbibitem 47:3, 47:4
\@ldots 30:511, 30:513
\@leftcolumn 54:123,
 54:2241, 54:2262, 54:2286, 54:2295

```

**File Key:** 01=ltidirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\@leftmark ..... 49:16, 49:79  
 \@let@token ..... 18:457, 18:458, 18:465, 32:83, 32:96, 05:49, 05:51, 38:256, 38:258, 38:261, 06:782, 06:785, 06:788, 06:796, 1228  
 \@align ..... 38:208, 38:210  
 \@linechar .. 42:190, 42:191, 42:192, 42:196, 42:197, 42:199, 42:204, 42:206, 42:207, 42:208, 42:209, 42:211, 42:215, 42:216, 42:219, 42:220, 42:225, 42:272, 42:669, 1240  
 \@linefnt .. 42:124, 42:127, 42:190, 42:265, 42:273, 42:304, 42:307, 42:676  
 \@linelen .. 42:164, 42:165, 42:176, 42:177, 42:203, 42:210, 42:219, 42:221, 42:226, 42:227, 42:228, 42:241, 42:242, 42:256, 42:257, 42:300, 42:303, 42:305, 42:306, 42:670  
 \@list ..... 803  
 \@listctr .. 39:226, 39:249, 47:9, 1196  
 \@listdepth ..... 39:23, 39:35, 39:38, 39:43, 39:99, 40:456, 802  
 \@listfiles ..... 20:62, 20:131, 20:186, 20:753, 20:772  
 \@listi ..... 803  
 \@listii ..... 803  
 \@listvi ..... 803  
 \@lnbk ..... 1234  
 \@loadwithoptions ..... 50:694, 50:716, 50:726, 50:735  
 \@lowpenalty ..... 18:118, 57:3  
 \@ltab ..... 41:71, 41:106  
 \@ltxnomath ..... 1209  
 \@m ..... 18:303, 18:336, 18:504, 18:509, 20:45, 20:115, 20:173, 39:80, 42:213, 42:217, 47:37, 02:21, 02:429, 02:431, 02:432, 02:498, 02:499, 382  
 \@mainaux ..... 20:3, 20:37, 20:38, 20:107, 20:108, 20:165, 20:166, 20:217, 20:299, 20:342, 20:361, 20:394, 20:408, 20:433, 37:22, 37:88, 37:150, 1220  
 \@makebox ..... 40:11, 40:20, 40:25  
 \@makecaption ..... 45:24, 1166  
 \@makecol 54:263, 54:415, 54:462, 54:484  
 \@makefcolumn ..... 54:395, 54:396, 54:404, 54:406, 54:442, 54:444, 54:452, 54:454, 54:2199, 54:2201, 54:2217, 54:2218  
 \@makefnmark ..... 45:400, 45:532  
 \@makefntext ..... 40:489, 40:506, 40:523, 45:476, 45:494, 45:512  
 \@makeother ..... 01:110, 24:431, 24:432, 24:433, 24:434, 24:435, 24:436, 24:437, 24:438, 24:439, 24:440, 24:441, 37:541, 37:562, 37:707, 37:722, 37:732, 50:438, 50:439, 50:1301, 50:1434, 50:1523, 06:800, 06:801, 01:60, 01:81  
 \@makepicbox ..... 40:10, 40:19, 40:67, 42:366, 42:424  
 \@makespecialcolbox ..... 54:507, 54:526, 1199  
 \@marbox ..... 45:320, 45:322, 45:326, 45:330, 45:331, 45:379, 54:1816, 54:1826, 54:1829, 54:1837, 54:1839, 54:1840, 54:1842, 54:1843, 54:1844, 54:1853  
 \@marginparreset 45:343, 45:361, 45:370  
 \@markright ..... 49:37, 49:62, 49:77  
 \@maxdepth .. 20:60, 20:129, 20:184, 54:91, 54:490, 54:518, 57:152, 1227  
 \@maxtab ..... 41:2, 41:94, 1210  
 \@medpenalty ..... 18:119, 57:3  
 \@meta@family@list ..... 29:116, 29:141, 29:153, 29:242, 29:506  
 \@midlist ..... 54:66, 54:503, 54:504, 54:1031, 54:1033, 54:1145, 54:1309, 54:1960, 54:1988  
 \@minipage... ..... 1232  
 \@minipagewhile ..... 39:205, 40:420, 40:422, 40:469, 45:187, 45:250, 45:345, 45:363  
 \@minipagerestore ..... 40:457, 40:459  
 \@minipagetrue ..... 40:421, 45:186  
 \@minus ..... 06:26, 54:2344, 54:2345, 54:2346, 54:2349, 54:2350  
 \@missing@onefilewithoptions ... 50:926, 50:982, 50:1100  
 \@missingfile@area ..... 20:646, 20:687, 20:700, 50:984  
 \@missingfile@base ..... 20:646, 20:688, 20:701, 50:985  
 \@missingfile@ext ..... 20:646, 20:689, 20:702, 50:986  
 \@missingfileerror 20:641, 20:656, 20:666, 20:675, 50:983, 50:1080, 461  
 \@mkboth ..... 49:11, 49:13  
 \@mklab ..... 39:45, 39:164  
 \@mkpream . 41:189, 41:234, 41:262, 1219  
 \@mparbottom ..... 45:387, 45:388, 54:120, 54:480, 54:1827, 54:1835, 54:1836, 54:1837, 54:1838  
 \@mpargs ..... 40:440, 40:472  
 \@mparswitchfalse ..... 54:104  
 \@mpfn ..... 40:454, 45:452, 45:457, 45:541, 45:545

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx, 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx, 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx, 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx, 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx, 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx, 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx, 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx, 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx, 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx, 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx, 56=ltphyphen.dtx, 57=ltfinal.dtx

\@mpfootins ..... 40:446, 40:463,  
   40:464, 40:467, 40:473, 40:480,  
   40:481, 40:498, 40:499, 40:515, 40:516  
 \@mpfootnotetext ..... 40:455, 40:475  
 \@mplistdepth ..... 40:456, 40:473  
 \@multicnt ..... 41:370,  
   41:372, 41:373, 41:374, 41:381,  
   41:382, 41:383, 42:103, 42:104,  
   42:106, 42:115, 42:116, 42:118,  
   42:666, 42:703, 42:705, 42:706,  
   42:707, 42:709, 42:710, 42:716,  
   42:722, 42:733, 42:737, 42:763,  
   42:765, 42:767, 42:769, 42:770,  
   42:774, 42:778, 42:789, 42:793, 1228  
 \@multiplelabels ... 20:33, 20:103,  
   20:161, 35:87, 35:93, 36:132, 37:60,  
   37:66, 37:125, 37:131, 37:164, 37:170  
 \@multiput ..... 42:86, 42:95, 42:98  
 \@multispan ..... 41:371, 41:375, 41:379  
 \@mypkg@name ..... 1027  
 \@mypkg@other@name ..... 1027  
 \@namedef .....  
   ... 20:446, 24:135, 24:136, 24:160,  
   25:5, 25:525, 26:418, 28:401, 28:405,  
   28:414, 33:57, 33:93, 33:866, 35:90,  
   37:287, 37:347, 37:364, 37:378,  
   37:584, 37:593, 38:426, 38:427,  
   06:50, 41:175, 43:12, 43:13, 43:18,  
   43:19, 43:23, 43:24, 43:25, 50:1214,  
   50:1349, 52:482, 57:480, 57:481, 1198  
 \@nameuse ..... 20:336, 20:392,  
   20:431, 20:445, 06:51, 43:23, 49:5,  
   50:920, 52:486, 54:611, 54:669, 1214  
 \@nbitem ..... 39:192, 39:245  
 \@ne ..... 02:16, 1214  
 \@needsf@rmat .. 50:762, 50:765, 50:770  
 \@needsformat .. 50:750, 50:760, 50:764  
 \@negargfalse ..... 42:186  
 \@negargtrue ..... 42:185  
 \@newcommand ..... 06:79, 06:80  
 \@newctr ..... 22:13, 22:15, 43:8  
 \@newenv ..... 06:150, 06:151, 06:160  
 \@newenva ..... 06:149, 06:148  
 \@newenvb ..... 06:151, 06:150  
 \@newfontswitch ..... 1208  
 \@newl@bel .....  
   35:84, 37:24, 37:90, 37:152, 47:10, 1226  
 \@newline ..... 18:95, 18:97  
 \@newlistfalse ..... 39:29,  
   39:33, 39:108, 39:206, 54:604, 54:662  
 \@newlisttrue .... 39:29, 39:33, 39:87  
 \@next ..... 45:60,  
   45:129, 45:319, 45:320, 54:9, 54:215,  
   54:313, 54:881, 54:901, 54:1816, 02:275  
 \@nextchar .....  
   41:269, 41:270, 41:330, 41:331, 41:332  
 \@nil ..... 03:13, 03:19, 03:83,  
   03:104, 03:105, 03:117, 03:118,  
   03:165, 03:166, 03:167, 13:13,  
   13:19, 13:27, 19:14, 20:247, 20:248,  
   20:249, 21:89, 21:110, 01:145,  
   01:146, 21:996, 21:1000, 21:1063,  
   21:1075, 21:1077, 24:383, 24:394,  
   24:510, 24:525, 24:629, 24:632,  
   24:633, 24:641, 25:431, 25:433,  
   25:465, 25:467, 25:645, 25:647,  
   25:671, 25:673, 26:350, 26:351,  
   26:353, 26:366, 26:372, 26:376,  
   26:377, 26:413, 26:434, 26:439,  
   26:519, 26:533, 27:26, 27:44, 27:53,  
   27:57, 28:40, 28:571, 28:579, 28:612,  
   28:1228, 28:1230, 32:58, 32:62,  
   06:53, 06:54, 06:58, 06:63, 41:367,  
   41:368, 06:135, 50:90, 50:91, 50:97,  
   50:105, 50:108, 50:115, 50:140,  
   50:191, 50:192, 50:203, 50:204,  
   50:214, 50:215, 50:217, 50:447,  
   50:470, 50:514, 50:520, 50:636,  
   50:656, 50:660, 50:666, 50:670,  
   50:877, 50:886, 50:903, 50:904,  
   06:451, 06:452, 50:1631, 50:1636,  
   50:1639, 50:1641, 50:1642, 50:1657,  
   50:1677, 50:1694, 50:1749, 50:1771,  
   50:1795, 52:168, 52:176, 52:369,  
   52:411, 52:413, 06:633, 06:776, 06:777  
 \@nbrlistfalse 39:33, 39:46, 39:91, 1196  
 \@nbrlisttrue ..... 39:249  
 \@nnil ..... 13:13, 13:20,  
   13:21, 13:22, 13:28, 20:248, 20:249,  
   24:214, 24:218, 24:219, 24:220,  
   24:235, 26:179, 26:181, 26:345,  
   26:347, 26:359, 26:361, 26:366,  
   26:380, 26:382, 26:389, 26:400,  
   26:401, 26:403, 26:434, 26:439,  
   42:13, 50:822, 50:823, 50:830,  
   50:850, 50:851, 50:858, 06:429, 06:435  
 \@no@font@optfalse .... 27:17, 27:129  
 \@no@lnbk ..... 18:9, 18:10, 18:41, 1234  
 \@no@pgbk ..... 18:7, 18:8, 18:33, 1227  
 \@nobreak ..... 943  
 \@nobreak... ..... 1232  
 \@nobreakfalse .....  
   ... 18:121, 18:123, 39:217,  
   44:94, 44:129, 44:157, 45:182,  
   54:167, 54:194, 54:1154, 54:1320, 1233  
 \@nobreaktrue ... 18:122, 44:126, 45:181  
 \@nocnterr ..... 14:240, 1207

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\@nocounterr . . . . . 14:240, 22:4,  
   22:8, 22:16, 22:110, 22:112, 43:21, 1208  
 \@nodocument . . . . . 14:245,  
   16:143, 20:68, 20:137, 20:192,  
   20:296, 20:358, 37:241, 45:39,  
   45:108, 54:158, 54:185, 54:214, 1233  
 \@noitemargfalse . . . . . 39:32, 39:224  
 \@noitemargtrue . . . . . 39:32, 39:167  
 \@noitemerr . . . . . . . . .  
   ... 14:280, 18:264, 18:281, 18:351,  
   18:373, 39:69, 39:81, 39:107, 1269  
 \@noitemerror . . . . . 1220  
 \@noligs . . . . . 37:542,  
   37:563, 37:723, 37:733, 37:744, 1231  
 \@nolnbk . . . . . 1234  
 \@nolnerr . . . . . 14:238,  
   18:43, 18:114, 37:417, 37:424, 1210  
 \@nomath 24:1, 24:337, 29:551, 29:586,  
   29:592, 29:613, 29:615, 29:637, 1194  
 \@noparitemfalse . . . . . 39:30, 39:169  
 \@noparitemtrue . . . . . 39:30, 39:66  
 \@noparlistfalse . . . . . 39:31, 39:70  
 \@noparlisttrue . . . . . 39:31, 39:67  
 \@nopgbk . . . . . 1227  
 \@normalcr . 18:53, 18:93, 40:419, 1250  
 \@normalsize . . . . . 50:4, 50:5, 1207  
 \@normalsize\_\check . . . . . 1207  
 \@noskipsec . . . . . 802  
 \@noskipsecfalse . . . . . 20:54,  
   20:124, 20:179, 44:98, 54:160, 54:187  
 \@noskipsectrue . . . . . 44:38, 44:95  
 \@notdefinable . . . . . 14:232,  
   06:136, 06:137, 06:141, 06:478, 06:519  
 \@notprerr 14:283, 20:66, 20:135, 20:190  
 \@nthm . . . . . 43:3, 43:4  
 \@nxtabmar . . . . . . . . .  
   . 41:11, 41:21, 41:23, 41:25, 41:75,  
   41:111, 41:112, 41:118, 41:119, 1229  
 \@obsoletefile . . . . . 20:738  
 \@oddfoot . . . . . 49:11,  
   49:14, 49:15, 54:126, 54:614, 54:673  
 \@oddhead . . . . . . . . .  
   . 49:11, 49:14, 54:125, 54:614, 54:673  
 \@onefilewithoptions 50:814, 50:818,  
   50:824, 50:842, 50:846, 50:852,  
   50:869, 50:873, 50:879, 50:895,  
   50:897, 50:980, 50:1047, 50:1600, 1260  
 \@onefilewithoptions@clashchk . . . . .  
   . . . . . 50:912, 50:962  
 \@onelevel@sanitize . . . . .  
   . . . . . 45:42, 45:111, 06:802, 1218  
 \@onfilewithoptions . . . . . 1006  
 \@onlypreamble . . . . . . . . .  
   . . . . . 08:2804, 20:197, 20:206,

20:243, 20:740, 20:800, 21:23, 21:24,  
   21:61, 21:62, 21:66, 21:125, 21:145,  
   21:189, 21:190, 21:204, 24:17,  
   24:115, 24:117, 24:123, 24:139,  
   24:167, 24:182, 24:203, 24:208,  
   24:250, 24:537, 26:419, 27:28,  
   27:36, 27:42, 27:79, 27:83, 27:88,  
   27:93, 27:98, 27:108, 27:126, 27:127,  
   27:128, 27:134, 27:138, 27:142,  
   28:17, 28:19, 28:44, 28:46, 28:107,  
   28:116, 28:136, 28:394, 28:395,  
   28:408, 28:454, 28:502, 28:514,  
   28:516, 28:529, 28:554, 28:611,  
   28:613, 28:655, 28:694, 28:710,  
   28:787, 28:881, 28:890, 28:946,  
   28:949, 28:952, 28:972, 28:985,  
   28:1039, 28:1074, 28:1085, 28:1099,  
   28:1153, 28:1173, 28:1177, 28:1241,  
   32:140, 32:141, 33:58, 33:59, 33:60,  
   33:867, 35:92, 06:66, 06:188,  
   06:190, 06:199, 06:207, 46:12,  
   46:29, 47:64, 47:81, 48:16, 48:364,  
   50:10, 50:13, 50:85, 50:112, 50:120,  
   50:122, 50:154, 50:357, 50:420,  
   50:426, 50:491, 50:494, 50:495,  
   50:506, 50:507, 50:508, 50:535,  
   50:541, 50:554, 50:591, 50:609,  
   50:629, 50:649, 50:673, 50:678,  
   50:682, 50:685, 50:693, 50:714,  
   50:717, 50:727, 50:746, 50:759,  
   50:764, 50:770, 50:798, 50:803,  
   50:891, 50:980, 50:1105, 50:1114,  
   50:1122, 50:1123, 50:1141, 50:1169,  
   50:1178, 50:1185, 50:1186, 50:1194,  
   50:1199, 50:1204, 50:1580, 50:1581,  
   50:1582, 50:1583, 50:1585, 51:199, 1258  
 \opargbegintheorem . . . . . 43:45, 43:48  
 \opcol . . . . . 54:264, 54:272, 54:396,  
   54:415, 54:444, 54:462, 54:467, 949  
 \options . . . . . 50:628  
 \othm . . . . . 43:3, 43:20  
 \outerparskip . . . . . . . . .  
   . 39:1, 39:88, 39:117, 39:176, 39:246  
 \outputbox . 48:382, 48:385, 48:389,  
   48:413, 48:416, 48:420, 54:122,  
   54:487, 54:489, 54:509, 54:512,  
   54:513, 54:533, 54:535, 54:536,  
   54:541, 54:544, 54:549, 54:551,  
   54:558, 54:564, 54:566, 54:640,  
   54:699, 54:727, 54:733, 54:743,  
   54:744, 54:767, 54:774, 54:860,  
   54:863, 54:866, 54:872, 54:873,  
   54:2241, 54:2245, 54:2246, 54:2260,  
   54:2266, 54:2286, 54:2292, 54:2301, 965

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

```

\@outputdblcol 54:471,
 54:2236, 54:2238, 54:2282, 54:2283
\@outputpage
 . 54:405, 54:454, 54:474, 54:594,
 54:2270, 54:2275, 54:2308, 54:2316
\@oval 42:453, 42:470
\@ovbtrue 42:475, 42:503, 42:533
\@ovdx 42:431, 42:486, 42:488, 42:494,
 42:496, 42:515, 42:517, 42:525,
 42:527, 42:542, 42:550, 42:552,
 42:588, 42:591, 42:598, 42:600,
 42:692, 42:693, 42:694, 42:695,
 42:711, 42:712, 42:713, 42:716,
 42:732, 42:752, 42:753, 42:754,
 42:755, 42:771, 42:772, 42:774, 42:788
\@ovdy 42:431, 42:487, 42:489, 42:495,
 42:496, 42:516, 42:518, 42:526,
 42:527, 42:543, 42:551, 42:552,
 42:563, 42:568, 42:576, 42:580,
 42:699, 42:700, 42:701, 42:702,
 42:717, 42:718, 42:719, 42:722,
 42:736, 42:759, 42:760, 42:761,
 42:762, 42:775, 42:776, 42:778, 42:792
\@ovhlinefalse 42:476, 42:504
\@ovhlinetrue 42:456, 42:460,
 42:464, 42:482, 42:488, 42:510, 42:517
\@ovhorz 42:493, 42:494,
 42:524, 42:525, 42:549, 42:550, 42:583
\@ovltrue 42:475, 42:503, 42:533
\@ovri 40:33, 42:431, 42:485, 42:514,
 42:541, 42:563, 42:576, 42:592, 42:601
\@ovro 42:431, 42:485, 42:494,
 42:495, 42:514, 42:525, 42:526,
 42:541, 42:550, 42:551, 42:562,
 42:568, 42:575, 42:580, 42:587,
 42:597, 42:613, 42:620, 42:629, 42:638
\@ovrtrue 42:475, 42:503, 42:533
\@ovttrue 42:475, 42:503, 42:533
\@ovvert 42:491, 42:492,
 42:520, 42:522, 42:545, 42:547, 42:556
\@ovvlinefalse 42:476, 42:504
\@ovvlinetrue
 . 42:459, 42:481, 42:489, 42:509, 42:518
\@ovxx 42:431, 42:479,
 42:481, 42:482, 42:486, 42:492,
 42:493, 42:507, 42:509, 42:510,
 42:515, 42:522, 42:524, 42:536,
 42:538, 42:542, 42:547, 42:549,
 42:587, 42:597, 42:689, 42:690,
 42:691, 42:695, 42:704, 42:705,
 42:714, 42:715, 42:716, 42:731,
 42:749, 42:750, 42:751, 42:755,
 42:764, 42:765, 42:773, 42:774, 42:787
\@covyy 42:431,
 42:480, 42:481, 42:482, 42:487,
 42:494, 42:508, 42:509, 42:510,
 42:516, 42:525, 42:537, 42:538,
 42:543, 42:550, 42:560, 42:573,
 42:696, 42:697, 42:698, 42:702,
 42:704, 42:720, 42:721, 42:722,
 42:735, 42:756, 42:757, 42:758,
 42:762, 42:764, 42:777, 42:778, 42:791
\@cp@filename 50:90,
 50:97, 50:105, 50:108, 50:115, 50:120
\@cp@filepath ... 50:91, 50:138, 50:147
\@cp@filepath@aux 50:139, 50:140, 50:148
\@pagedp ... 54:119, 54:310, 54:315,
 54:1095, 54:1248, 54:1845, 54:1855
\@pageht ... 54:118, 54:311, 54:315,
 54:317, 54:318, 54:319, 54:323,
 54:1094, 54:1247, 54:1828, 54:1835
\@par 15:3, 15:5
\@parboxrestore ... 40:334, 40:361,
 40:419, 40:453, 40:484, 40:502,
 40:519, 45:19, 45:100, 45:169,
 45:342, 45:360, 45:470, 45:489,
 45:507, 54:221, 54:605, 54:663, 1233
\@parboxto 40:325, 1241
\@parmoderr 14:273, 45:58, 45:127, 45:316
\@parse@version 03:83,
 03:104, 03:105, 03:117, 03:118,
 03:165, 03:166, 03:167, 50:197,
 50:203, 50:204, 50:214, 50:1642,
 50:1657, 50:1694, 50:1771, 50:1795
\@parse@version@
 . 50:191, 50:192, 50:197, 50:209
\@parse@version@dash
 . 50:215, 50:217, 1246
\@partaux 20:3, 20:223,
 20:267, 20:285, 20:309, 20:311,
 20:312, 20:332, 20:371, 20:373,
 20:374, 20:388, 20:418, 20:420,
 20:421, 20:427, 20:439, 20:441, 20:444
\@partlist
 . 20:232, 20:236, 20:237, 20:239,
 20:263, 20:282, 20:305, 20:367, 20:414
\@partsfalse 20:6
\@partstrue ... 20:231, 20:261, 20:281
\@pass@ptions
 . 50:443, 50:445, 50:465, 50:466,
 50:468, 50:481, 50:482, 50:484,
 50:491, 50:492, 50:493, 50:991, 50:1073
\@pboxswfalse .. 40:332, 40:359, 40:438
\@pboxswtrue 40:342, 40:369
\@pdef 1203
\@penup 38:199, 38:200

```

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\@percentchar ... 50:1293, 50:1295,  
   50:1297, 50:1299, 50:1426, 50:1428,  
   50:1430, 50:1432, 50:1515, 50:1517,  
   50:1519, 50:1521, 50:1569, 01:90  
 \@pgbk ..... 1227  
 \@picbox .....  
   . 42:6, 42:31, 42:43, 42:51, 42:52, 1195  
 \@picht ..... 42:6, 42:29, 42:42, 42:51  
 \@picture ..... 42:23, 42:24  
 \@picture@warn .....  
   . 42:223, 42:441, 42:445, 42:449  
 \@pkgetension ..... 1203  
 \@pkgextension 50:31, 50:155, 50:163,  
   50:221, 50:492, 50:684, 50:687,  
   50:726, 50:735, 50:825, 50:853,  
   50:880, 50:1002, 50:1029, 50:1150,  
   50:1180, 50:1614, 52:482, 52:492, 1004  
 \@plus ..... 18:532,  
   06:26, 44:16, 44:206, 44:229,  
   49:83, 54:2344, 54:2345, 54:2346,  
   54:2349, 54:2350, 54:2354, 54:2355,  
   54:2356, 54:2360, 54:2361, 54:2362  
 \@pnumwidth ..... 44:218, 44:241  
 \@popfilename 50:33, 50:960, 50:1095, 981  
 \@pr@videopackage .....  
   . 50:356, 50:370, 50:373,  
   50:394, 50:396, 50:407, 50:409, 50:420  
 \@preamble .....  
   . 41:190, 41:192, 41:200, 41:237,  
   41:256, 41:258, 41:259, 41:263,  
   41:278, 41:296, 41:297, 41:334, 55:129  
 \@preamblecmds ..... 20:67, 20:136,  
   20:191, 06:66, 50:1809, 50:1810, 940  
 \@preamerr 14:260, 41:199, 41:274, 41:355  
 \@process@pti@ns .....  
   . 50:553, 50:571, 50:588,  
   50:595, 50:596, 50:609, 50:613, 50:615  
 \@process@ptions 50:540, 50:542, 50:554  
 \@protect@... ..... 1214  
 \@protected@testopt .....  
   . 06:89, 06:101, 06:628, 06:640  
 \@protecteddef ..... 1209  
 \@providesfile .....  
   . 50:430, 57:704, 01:82, 01:83  
 \@optionlist .....  
   . 50:152, 50:224, 50:539, 50:968,  
   50:974, 50:1063, 50:1069, 50:1181  
 \@pushfilename .....  
   . 50:33, 50:905, 50:1050, 980  
 \@put ..... 42:452,  
   42:496, 42:527, 42:552, 42:620, 42:638  
 \@qend ..... 14:236, 06:136, 06:776  
 \@qrelax ..... 06:137, 06:776

\@raw@classoptionslist .....  
   . 50:11, 50:812, 51:97, 51:139  
 \@rc@ifdefinable .....  
   . 21:14, 06:130, 06:132, 06:250  
 \@reargdef ..... 06:122, 1203  
 \@refundefined 20:55, 20:125, 20:180,  
   35:3, 37:58, 37:123, 37:162, 1229  
 \@reinserts ..... 54:329, 54:332, 54:520  
 \@remove@eq@value 50:509, 50:576, 50:636  
 \@remove@tlig ..... 21:996, 21:1004  
 \@remove@tlig@ ..... 21:996, 21:997  
 \@remove@tlig@@ ..... 21:997, 21:1000  
 \@remove@tlig@@@ ..... 21:1010, 21:1029  
 \@removeelement . 13:32, 50:635, 50:644  
 \@removefromreset ..... 22:94,  
   22:118, 22:119, 22:129, 22:152, 22:155  
 \@renewfontswitch ..... 1203  
 \@reqcolroom ..... 54:1094, 54:1095,  
   54:1098, 54:1100, 54:1101, 54:1106,  
   54:1110, 54:1112, 54:1140, 54:1141,  
   54:1247, 54:1248, 54:1252, 54:1255,  
   54:1256, 54:1261, 54:1268, 54:1270,  
   54:1302, 54:1303, 54:1414, 54:1416,  
   54:1418, 54:1421, 54:1423, 54:1497,  
   54:1500, 54:1503, 54:1508, 54:1510,  
   54:2024, 54:2141, 54:2146, 54:2149  
 \@reserved@ ..... 562  
 \@reset@ptions ..... 50:924, 50:961,  
   50:989, 50:1054, 50:1096, 50:1106  
 \@resetactivechars .....  
   . 54:579, 54:602, 54:660  
 \@resethfps .....  
   . 54:1209, 54:1378, 54:2091, 1210  
 \@resetprotect ..... 1224  
 \@restorepar . 15:5, 18:388, 18:404,  
   18:422, 18:438, 39:127, 39:137, 1226  
 \@reversemarginfalse . 45:388, 54:103  
 \@reversemargintrue ..... 45:387  
 \@rightmark ..... 49:16, 49:80  
 \@rightskip ..... 37:456,  
   37:475, 37:493, 39:75, 40:392, 40:413  
 \@rjfieldfalse ..... 41:34, 41:77  
 \@rjfieldtrue ..... 41:125  
 \@rmfamilyhook .....  
   . 29:429, 29:451, 29:463, 29:490  
 \@roman ..... 22:185, 22:191  
 \@rsbox ..... 40:562, 40:569, 40:573  
 \@rtab ..... 41:71, 41:86  
 \@rule ..... 40:534, 40:539, 40:543  
 \@sanitize .....  
   . 46:7, 46:18, 46:24, 46:35, 06:800  
 \@savebox ..... 40:161, 40:168, 40:174  
 \@savemarbox .....  
   . 45:326, 45:327, 45:330, 45:333

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@savepicbox . . . 40:161, 40:168, 40:178
\@savsf 17:28, 17:30,
 17:50, 17:52, 18:124, 18:130, 18:139,
 18:158, 18:172, 18:184, 18:198, 18:212
\@savsk . . . 18:124, 18:129, 18:140,
 18:159, 18:173, 18:185, 18:199, 18:213
\@scolelt 54:792, 54:857
\@sdblcolelt . . . 54:809, 54:829, 54:858
\@secCntformat . . . 44:60, 44:111, 1197
\@secondoffive 35:24, 35:27, 35:41, 35:58
\@secondoftwo 03:87, 03:185,
 17:5, 20:509, 20:539, 20:585, 20:602,
 21:131, 22:238, 22:243, 29:512,
 33:71, 33:836, 33:886, 33:902, 35:44,
 05:36, 35:80, 05:40, 06:212, 49:17,
 50:159, 50:193, 50:205, 50:238,
 50:256, 50:305, 50:335, 06:457,
 50:1800, 06:586, 06:637, 06:745,
 06:752, 06:772, 06:819, 57:710, 01:72
\@secpenalty 18:18, 44:36, 44:50
\@sect 44:54, 44:55
\@seqnrcr 38:425
\@set@curr@file@aux . . . 52:363, 1264
\@setckpt 20:439,
 20:446, 37:23, 37:89, 37:151, 1227
\@setfloattypecounts
 54:1078, 54:1228, 54:1399,
 54:1480, 54:1575, 54:1696, 54:2038
\@setfontsize 29:637
\@setfps 45:34
\@setfpsbit 45:73, 45:75, 45:77,
 45:85, 45:143, 45:146, 45:149, 54:2082
\@setmarks . . . 54:2252, 54:2254, 54:2269
\@setminipage 40:458,
 45:21, 45:177, 45:185, 45:376, 1221
\@setminipage 1234
\@setnobreak 45:179, 45:375, 1221
\@setpar 15:3, 39:78
\@setprotect 1224
\@setref 35:10
\@setspace 29:637
\@settab 41:71, 41:93
\@settodim 23:17
\@settopoint 23:35
\@setupverbvisiblespace . . . 37:585,
 37:597, 37:601, 37:662, 37:676, 37:690
\@setupverbvisibletab . . . 37:613, 37:634
\@sffamilyhook
 29:429, 29:456, 29:464, 29:491
\@sharp 41:196, 41:235, 41:265, 41:280,
 41:281, 41:301, 41:303, 41:305, 41:333
\@shipoutsetup 54:594
\@shortstack 42:132, 42:133
\@show@. 100
\@show@DeclareRobustCommand
 06:554, 06:568, 06:613, 06:667, 06:670
\@show@DeclareRobustCommand@env . . .
 06:691, 06:693
\@show@environment . . . 06:685, 06:686
\@show@environment@begin . . .
 06:694, 06:700, 06:708, 06:713
\@show@environment@end . . .
 06:698, 06:707, 06:712, 105
\@show@environment@end@aux . . .
 06:716, 06:718
\@show@macro 06:613, 06:703
\@show@newcommand . . . 06:555, 06:568,
 06:621, 06:648, 06:667, 06:673, 102
\@show@newcommand@aux . . .
 06:648, 06:677, 06:695, 103
\@show@newcommand@env . . . 06:693, 06:692
\@show@nonstop 06:703, 06:724, 105
\@show@normalenv 06:689, 06:712
\@show@tokens . . .
 06:653, 06:658, 06:702, 06:719
\@show@typeout . . .
 06:697, 06:702, 06:724, 105
\@showcommandlisthook . . .
 06:551, 06:553, 06:561, 07:1438, 100
\@showenvironmentlisthook . . .
 06:688, 06:690, 07:1440, 1264
\@skipping@modulefalse . . .
 03:154, 03:173, 03:200
\@skipping@moduletrue . . . 03:153, 03:177
\@sline . . . 42:167, 42:179, 42:184, 42:269
\@slowromancap 22:192, 22:193
\@spaces 14:218
\@specialoutput 54:258
\@specialpagefalse 54:99, 54:611, 54:669
\@specialpagetrue 49:9
\@specialstyle . . . 49:9, 54:611, 54:669
\@spoken 06:785, 06:795
\@sqrt 38:355
\@sect 44:53, 44:112
\@stackcr 42:139, 42:142
\@star@or@long 06:72, 06:77, 06:124,
 06:146, 06:152, 06:178, 06:187, 06:233
\@startcolumn . . . 54:265, 54:272, 54:779
\@startdblcolumn 54:779,
 54:2274, 54:2277, 54:2314, 54:2320
\@startfield 41:28,
 41:46, 41:92, 41:104, 41:125, 41:133
\@startline . . .
 41:20, 41:57, 41:62, 41:68, 41:83, 41:154
\@startpagehook 1200
\@startpbox 41:193,
 41:236, 41:266, 41:332, 41:384, 41:386
\@startsection 44:39

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\@starttoc 44:149, 1164
\@stopfield 41:32, 41:48, 41:86, 41:93,
 41:125, 41:127, 41:140, 41:142, 41:154
\@stopline 41:30, 41:56, 41:85
\@stpelt 22:20, 22:23
\@string@makeletter 06:806, 109
\@strip@args 21:76
\@strip@tex@ext 20:227,
 20:237, 20:239, 20:244, 20:274, 450
\@strip@tex@ext@aux .. 20:244, 20:275
\@svector 42:244, 42:259, 42:269
\@sverb 37:646, 37:725, 37:734, 37:737, 781
\@svsec 44:57, 44:60, 44:66, 44:78, 1197
\@svsechd 44:76, 44:101, 44:121
\@swaptwoargs 20:619,
 20:621, 50:934, 52:153, 52:167, 52:187
\@sxverbatim ... 37:518, 37:586, 37:593
\@tabacckludge 21:223, 21:225, 21:454,
 21:455, 33:193, 33:200, 33:202, 1251
\@tabacol 41:178, 41:258
\@tabarray 41:170, 41:180, 41:181
\@tabclassiv 41:180, 41:330
\@tabclassz 41:179, 41:282
\@tabcr 41:56, 41:73
\@tabfbox 41:16, 41:80, 41:82
\@tablab 41:72, 41:126
\@tabminus 41:72, 41:117
\@tabplus 41:72, 41:110
\@tabpush 41:11,
 41:77, 41:85, 41:140, 41:143, 41:145
\@tabrj 41:72, 41:124
\@tabular 41:174, 41:177, 41:178
\@tabularcr 41:180, 41:208
\@tempa 35:151, 35:152, 35:153, 35:154,
 35:178, 35:179, 35:189, 35:190, 1221
\@tempboxa 12:13, 21:69,
 23:21, 23:22, 23:23, 23:28, 23:29,
 39:229, 39:235, 39:236, 39:238,
 40:29, 40:30, 40:31, 40:32, 40:37,
 40:38, 40:39, 40:40, 40:214, 40:248,
 40:267, 40:276, 40:286, 40:441,
 40:472, 40:579, 40:580, 40:581,
 40:588, 40:589, 40:590, 40:591,
 42:304, 42:305, 42:447, 42:448,
 42:485, 42:490, 42:495, 42:496,
 42:514, 42:519, 42:526, 42:527,
 42:541, 42:544, 42:551, 42:552,
 42:613, 42:614, 42:619, 42:620,
 42:629, 42:630, 42:637, 42:638,
 42:723, 42:741, 42:779, 42:797,
 44:138, 44:139, 45:322, 45:380,
 54:307, 54:379, 54:384, 54:385,
 54:426, 54:431, 54:432, 54:570,
 54:630, 54:637, 54:638, 54:689,
 54:696, 54:697, 54:725, 54:729,
 54:741, 54:747, 54:754, 54:755,
 54:756, 54:757, 54:761, 54:769, 1196
\@tempcnta 12:7, 28:954, 28:955, 28:956,
 28:957, 28:961, 41:242, 41:243,
 41:244, 41:245, 42:187, 42:188,
 42:214, 42:215, 42:216, 42:229,
 42:230, 42:231, 42:232, 42:239,
 42:240, 42:254, 42:255, 42:270,
 42:271, 42:276, 42:278, 42:279,
 42:280, 42:281, 42:282, 42:285,
 42:287, 42:288, 42:289, 42:290,
 42:291, 42:292, 42:293, 42:294,
 42:295, 42:296, 42:335, 42:336,
 42:337, 42:338, 42:339, 42:360,
 42:361, 42:362, 42:363, 42:364,
 42:365, 42:392, 42:393, 42:394,
 42:395, 42:396, 42:414, 42:416,
 42:418, 42:419, 42:421, 42:423,
 42:438, 42:439, 42:440, 42:442,
 42:444, 42:446, 42:448, 42:561,
 42:566, 42:574, 42:578, 42:615,
 42:616, 42:617, 42:618, 42:631,
 42:632, 42:634, 42:635, 42:657,
 42:658, 42:659, 42:660, 42:661,
 42:662, 42:710, 42:730, 42:770,
 42:786, 45:62, 45:68, 45:70, 45:79,
 45:80, 45:90, 45:91, 45:131, 45:137,
 45:139, 45:152, 45:153, 45:159,
 45:160, 50:1241, 50:1243, 50:1244,
 50:1245, 50:1372, 50:1374, 50:1375,
 50:1376, 50:1486, 50:1488, 50:1489,
 50:1490, 54:16, 54:18, 54:20,
 54:938, 54:939, 54:940, 54:941,
 54:961, 54:962, 54:963, 54:964,
 54:986, 54:989, 54:1022, 54:1025,
 54:1136, 54:1298, 54:1578, 54:1581,
 54:1699, 54:1702, 54:1817, 54:1819,
 54:1822, 54:1824, 54:1826, 54:1848,
 54:2072, 54:2073, 54:2077, 54:2083,
 54:2087, 57:227, 57:232, 57:233,
 57:234, 57:364, 57:366, 57:367,
 57:368, 57:375, 57:377, 57:378,
 57:379, 57:385, 57:387, 57:388,
 57:389, 57:396, 57:398, 57:399,
 57:400, 57:426, 57:428, 57:429,
 57:430, 57:452, 57:454, 57:455,
 57:456, 57:462, 57:464, 57:465,
 57:466, 57:490, 57:495, 57:496, 57:497
\@tempcntb ... 12:7, 28:955, 28:959,
 28:961, 42:279, 42:280, 42:281,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,
56=ltphypen.dtx, 57=ltfinal.dtx

42:283, 42:284, 42:285, 42:561,  
 42:562, 42:566, 42:567, 42:574,  
 42:575, 42:578, 42:579, 45:88, 45:89,  
 45:90, 45:157, 45:158, 45:159, 54:17,  
 54:20, 54:21, 54:2083, 54:2084,  
 54:2085, 57:228, 57:232, 57:491, 57:495  
 \@tempdima ..... 12:10,  
 24:219, 24:224, 38:186, 38:189,  
 38:195, 40:47, 40:48, 40:62, 40:63,  
 40:247, 40:248, 40:266, 40:267,  
 40:274, 40:275, 40:276, 40:278,  
 40:333, 40:334, 40:360, 40:361,  
 40:439, 40:443, 40:546, 40:549,  
 40:550, 40:577, 40:579, 40:585,  
 40:588, 41:35, 41:36, 41:37, 41:88,  
 41:89, 41:90, 41:91, 41:218, 41:219,  
 42:210, 42:211, 42:213, 42:214,  
 42:215, 42:216, 42:217, 42:218,  
 42:437, 42:438, 42:439, 42:448,  
 42:486, 42:487, 42:491, 42:492,  
 42:515, 42:516, 42:520, 42:522,  
 42:542, 42:543, 42:545, 42:547,  
 42:616, 42:618, 42:633, 42:635,  
 42:636, 42:656, 42:657, 42:658,  
 44:211, 44:212, 44:234, 44:235,  
 44:248, 45:196, 45:198, 45:218,  
 45:220, 45:258, 45:259, 45:260,  
 54:231, 54:232, 54:233, 54:491,  
 54:493, 54:539, 54:541, 54:542,  
 54:547, 54:552, 54:556, 54:561,  
 54:565, 54:921, 54:924, 54:944,  
 54:954, 54:966, 54:976, 54:1641,  
 54:1642, 54:1645, 54:1646, 54:1766,  
 54:1767, 54:1771, 54:1772, 54:1827,  
 54:1828, 54:1829, 54:1830, 54:1833,  
 54:1836, 54:1839, 54:1841, 54:2190,  
 54:2191, 54:2193, 54:2194, 1197  
 \@tempdimb ..... 12:10,  
 24:220, 24:225, 24:649, 24:653,  
 26:179, 26:180, 26:437, 26:460,  
 26:461, 26:470, 26:471, 26:475,  
 26:493, 26:496, 26:499, 26:501,  
 40:336, 40:337, 40:363, 40:364,  
 40:547, 40:550, 40:578, 40:580,  
 40:586, 40:589, 42:211, 42:212,  
 42:357, 42:359, 42:362, 42:365,  
 42:481, 42:483, 42:484, 42:509,  
 42:512, 42:513, 42:538, 42:539,  
 42:540, 42:611, 42:612, 42:621,  
 42:627, 42:628, 42:639, 42:647,  
 42:648, 42:653, 54:944, 54:945,  
 54:946, 54:947, 54:954, 54:966,  
 54:967, 54:968, 54:969, 54:976, 1197  
 \@tempdimc .. 12:10, 26:454, 26:455,  
 26:457, 26:458, 26:460, 26:461,  
 40:74, 40:75, 40:85, 40:86, 40:548,  
 40:549, 40:550, 42:30, 42:31, 42:32,  
 42:33, 42:34, 42:35, 42:60, 42:61,  
 42:63, 42:64, 42:332, 42:333, 42:334  
 \@tempdimx ..... 1197  
 \@tempskipa ..... 12:14, 18:45,  
 18:48, 18:49, 18:300, 18:307, 18:309,  
 18:312, 18:333, 18:340, 18:342,  
 18:345, 26:181, 26:182, 39:116,  
 39:117, 39:118, 39:174, 39:176,  
 39:177, 39:178, 39:246, 39:247,  
 39:248, 44:42, 44:44, 44:45, 44:50,  
 44:62, 44:63, 44:88, 44:89, 44:91,  
 44:103, 44:104, 44:113, 44:114,  
 54:1876, 54:1877, 54:1879, 54:1887  
 \@tempskipb ..... 12:14,  
 18:221, 18:223, 18:225, 18:228,  
 18:230, 18:244, 18:259, 18:276,  
 18:298, 18:300, 18:301, 18:305,  
 18:307, 18:309, 18:310, 18:331,  
 18:333, 18:334, 18:338, 18:340,  
 18:342, 18:343, 18:365, 18:368, 433  
 \@tempswa ..... 1193  
 \@tempswafalse ... 20:303, 20:365,  
 20:412, 21:1537, 24:94, 24:699,  
 28:466, 28:541, 28:615, 28:696,  
 28:1216, 28:1222, 37:26, 37:91,  
 37:153, 37:533, 37:554, 47:17, 47:29,  
 50:1214, 50:1230, 50:1349, 50:1365,  
 52:286, 52:309, 52:330, 54:992,  
 54:1028, 54:1584, 54:1705, 02:262, 01:62  
 \@tempswatrue 20:301, 20:306, 20:363,  
 20:368, 20:410, 20:415, 21:1540,  
 21:1541, 24:97, 24:700, 24:701,  
 24:704, 24:707, 28:469, 28:544,  
 28:618, 28:699, 28:1179, 36:152,  
 37:196, 37:538, 37:559, 47:17, 47:29,  
 50:1211, 50:1346, 52:286, 52:309,  
 52:330, 54:1586, 54:1609, 54:1707,  
 54:1732, 54:2151, 54:2168, 02:268, 01:63  
 \@temptokena ..... 12:16, 37:208,  
 37:212, 37:221, 37:234, 37:235,  
 49:27, 49:31, 49:38, 49:42, 49:55,  
 49:56, 49:63, 49:64, 49:77, 49:78, 767  
 \@test... ..... 1232  
 \@test@opt ..... 1228  
 \@testdef .. 37:24, 37:90, 37:152, 37:194  
 \@testfalse ..... 54:12, 54:14, 54:15  
 \@testfp ... 54:886, 54:906, 54:942,  
 54:965, 54:2075, 54:2204, 54:2221, 1220  
 \@testopt ..... 18:7,  
 18:8, 18:9, 18:10, 38:396, 06:33,  
 06:79, 06:99, 06:103, 06:148, 1228

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

\@testpach ..... 41:270, 41:348  
 \@testtrue ..... 54:13, 54:21, 54:358,  
     54:889, 54:908, 54:948, 54:970, 54:2079  
 \@testwrongwidth .. 54:347, 54:887,  
     54:943, 54:1116, 54:1430, 54:1635  
 \@text@composite .....  
     ..... 21:76, 21:1070, 21:1075, 1222  
 \@text@composite@x ..... 21:76  
 \@textbottom ..... 49:83,  
     49:85, 54:515, 54:553, 54:567, 54:576  
 \@textfloatsheight .....  
     54:480, 54:1091, 54:1093, 54:1143,  
     54:1144, 54:1149, 54:1244, 54:1246,  
     54:1306, 54:1308, 54:1314, 54:2024  
 \@textmin .....  
     ... 45:285, 45:286, 45:299, 45:300,  
     54:114, 54:1093, 54:1097, 54:1100,  
     54:1101, 54:1246, 54:1251, 54:1255,  
     54:1256, 54:1418, 54:1503, 54:1602,  
     54:1604, 54:1620, 54:1724, 54:1726,  
     54:1744, 54:2132, 54:2134, 54:2136  
 \@textsubscript ..... 45:424, 45:432  
 \@textsupserscript .....  
     ..... 45:401, 45:403, 45:404, 1225  
 \@texttop .....  
     ... 49:83, 49:85, 54:511, 54:534, 54:576  
 \@tf@r ..... 13:25, 13:26, 1206  
 \@tf@r ..... 13:25, 20:586, 20:603,  
     20:755, 32:88, 40:78, 40:97, 41:268,  
     42:477, 42:505, 42:534, 45:63, 45:132  
 \@tf@rloop ..... 13:27, 13:28, 13:30  
 \@thanks ..... 44:11, 44:34  
 \@thefnmark ..... 40:487, 40:504,  
     40:521, 45:400, 45:401, 45:453,  
     45:458, 45:473, 45:491, 45:509,  
     45:521, 45:526, 45:537, 45:542, 45:553  
 \@thefoot ..... 54:126, 54:614,  
     54:617, 54:644, 54:673, 54:676, 54:703  
 \@thehead ..... 54:125, 54:614,  
     54:616, 54:634, 54:673, 54:675, 54:693  
 \@themargin ..... 54:74, 54:615,  
     54:617, 54:629, 54:674, 54:676, 54:688  
 \@themark .....  
     ... 49:26, 49:27, 49:37, 49:38, 49:41,  
     49:54, 49:55, 49:62, 49:63, 49:78, 49:81  
 \@thirdofthree ..... 21:197, 06:216  
 \@thm ..... 43:12, 43:18, 43:24, 43:26  
 \@thmcounter ..... 43:11, 43:17, 43:46  
 \@thmcountersep ..... 43:10, 43:46  
 \@title ..... 44:7, 44:31  
 \@tocrmarg ..... 44:207, 44:230  
 \@toodeep . 14:253, 39:36, 39:256, 39:267  
 \@toplist ..... 54:64,  
     54:386, 54:387, 54:433, 54:434,  
     54:720, 54:726, 54:736, 54:737,  
     54:1029, 54:1041, 54:1958, 54:1986  
 \@topnewpage ..... 54:201, 1209  
 \@topnum ..... 45:271,  
     54:107, 54:1026, 54:1027, 54:1041,  
     54:1045, 54:1462, 54:1467, 54:1555,  
     54:1562, 54:1949, 54:1977, 54:2018  
 \@toproom ..... 45:273, 54:108,  
     54:1029, 54:1041, 54:1950, 54:1978  
 \@topsep ... 39:1, 39:71, 39:73, 39:195  
 \@topsepadd .....  
     ... 39:1, 39:59, 39:61, 39:71, 39:124  
 \@totallftmargin .....  
     37:529, 37:551, 39:9, 39:53, 39:54,  
     40:391, 40:412, 41:35, 41:76, 41:81, 802  
 \@trivlist ..... 39:48, 39:57, 39:92  
 \@tryfcolumn ..... 54:782,  
     54:802, 54:820, 54:836, 54:2205, 54:2222  
 \@trylist .....  
     54:845, 54:848, 54:881, 54:901, 54:923  
 \@ttfamilyhook .....  
     ... 29:429, 29:461, 29:465, 29:492  
 \@twoclasseserror ..... 50:675, 50:1200  
 \@twocolumnfalse ..... 54:101, 54:149  
 \@twocolumntrue ..... 54:208  
 \@twoheadclasserror .....  
     ... 50:959, 50:1094, 50:1195  
 \@twosidefalse ..... 54:102  
 \@typein 06:32, 06:33, 06:40, 06:48, 1225  
 \@typeset@protect .....  
     ... 21:26, 21:32, 21:210, 21:218,  
     29:638, 37:280, 37:317, 06:102,  
     06:255, 06:262, 06:264, 57:358, 1220  
 \@cuclclist .....  
     21:1506, 21:1507, 21:1564, 57:571, 1235  
 \@undefined .....  
     42:57, 42:71, 42:82, 42:91, 42:162,  
     42:174, 42:237, 42:252, 42:313, 42:371  
 \@undefined ..... 01:92, 02:582,  
     01:93, 01:94, 02:650, 02:690, 02:705,  
     10:182, 10:183, 10:184, 10:186,  
     10:187, 10:188, 10:190, 10:191,  
     03:53, 03:62, 01:115, 14:28, 01:123,  
     16:170, 16:171, 16:172, 16:173,  
     16:174, 17:61, 04:2, 18:90, 04:15,  
     04:16, 04:17, 04:29, 04:30, 18:497,  
     18:546, 18:552, 01:131, 20:61, 20:62,  
     04:74, 20:130, 20:131, 20:185,  
     20:186, 04:84, 20:274, 20:275,  
     20:477, 20:478, 20:479, 20:484,  
     20:538, 20:551, 20:560, 20:575,  
     20:630, 01:138, 21:195, 21:197,  
     04:173, 21:333, 21:335, 21:337,  
     21:339, 21:341, 21:343, 21:345,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltt@rror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspac@.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

21:347, 21:349, 21:351, 21:370, 33:736, 33:737, 33:738, 33:739,  
 21:372, 21:374, 04:189, 21:458, 33:740, 33:741, 33:742, 33:743,  
 04:197, 04:205, 21:700, 21:703, 33:744, 33:745, 33:746, 33:747,  
 04:237, 04:238, 04:239, 04:240, 33:748, 33:749, 33:750, 33:751,  
 04:241, 04:242, 04:243, 04:244, 33:752, 33:753, 33:754, 33:755,  
 04:245, 04:246, 04:247, 04:248, 33:756, 33:757, 33:762, 33:763,  
 04:249, 04:250, 04:251, 04:252, 33:765, 33:766, 33:767, 33:768,  
 04:253, 04:254, 04:255, 04:261, 33:769, 33:770, 33:771, 33:772,  
 22:252, 24:459, 24:499, 24:561, 33:774, 33:775, 33:777, 33:778,  
 24:594, 24:658, 24:665, 25:388, 33:780, 33:781, 33:782, 33:783,  
 25:411, 25:420, 25:509, 25:510, 33:785, 33:786, 33:787, 33:789,  
 25:511, 25:512, 25:513, 25:514, 33:791, 33:792, 33:793, 33:794,  
 25:515, 25:516, 25:517, 25:518, 33:795, 33:796, 33:797, 33:799,  
 25:529, 25:534, 25:539, 25:603, 33:800, 33:801, 33:802, 33:803,  
 25:604, 25:605, 25:606, 25:607, 33:804, 33:805, 33:806, 33:807, 05:4,  
 25:608, 25:609, 25:634, 25:714, 01:226, 05:13, 05:14, 05:15, 05:16,  
 25:716, 25:717, 25:719, 25:720, 35:161, 35:162, 35:163, 35:183,  
 25:721, 25:723, 26:552, 26:553, 27:4, 35:199, 35:200, 36:267, 36:268,  
 27:5, 27:6, 27:7, 27:8, 27:9, 27:10, 36:270, 36:271, 36:272, 36:274,  
 27:11, 27:12, 27:13, 27:14, 27:15, 36:275, 36:276, 36:278, 36:279,  
 27:16, 27:17, 27:18, 27:19, 27:20, 37:176, 37:228, 37:229, 37:230,  
 28:141, 28:509, 28:561, 01:189, 37:231, 37:306, 37:385, 37:405,  
 28:812, 28:919, 01:193, 29:35, 37:406, 37:407, 37:408, 37:515,  
 29:58, 29:71, 29:91, 29:104, 29:105, 05:150, 05:151, 05:152, 05:153,  
 29:106, 29:107, 29:108, 29:109, 05:154, 37:577, 05:155, 37:595,  
 29:110, 29:111, 29:112, 29:113, 37:596, 37:597, 37:598, 37:631,  
 29:114, 29:116, 29:117, 29:118, 37:643, 37:692, 05:172, 05:173,  
 29:231, 29:232, 29:233, 29:234, 05:174, 05:175, 38:236, 38:246,  
 29:235, 29:236, 29:237, 29:238, 38:247, 38:283, 38:286, 38:329,  
 29:239, 29:240, 29:242, 29:243, 38:342, 05:195, 38:436, 05:196,  
 29:244, 29:304, 29:305, 29:472, 40:21, 40:169, 40:233, 40:314,  
 29:490, 29:491, 29:492, 29:534, 06:34, 40:540, 40:570, 42:18, 42:466,  
 29:535, 29:536, 29:537, 29:538, 42:467, 44:197, 44:245, 45:5, 06:235,  
 29:578, 29:579, 29:580, 29:581, 45:429, 45:448, 45:562, 47:30, 47:53,  
 29:582, 29:583, 29:594, 29:716, 06:319, 06:320, 50:4, 50:25, 06:351,  
 30:15, 30:51, 30:58, 30:73, 32:37, 50:146, 50:147, 50:148, 50:209,  
 32:38, 32:39, 32:122, 33:159, 33:367, 06:379, 06:383, 06:384, 50:416,  
 33:617, 33:620, 33:621, 01:219, 06:401, 50:526, 50:576, 06:415,  
 33:651, 33:652, 33:653, 33:656, 50:981, 50:1019, 50:1021, 50:1027,  
 33:657, 33:658, 33:659, 33:660, 50:1084, 50:1099, 50:1100, 50:1115,  
 33:661, 33:666, 33:667, 33:668, 06:462, 06:463, 50:1254, 06:464,  
 33:669, 33:674, 33:675, 33:676, 06:465, 06:466, 06:467, 06:468,  
 33:677, 33:680, 33:681, 33:682, 50:1332, 50:1335, 50:1385, 50:1464,  
 33:684, 33:685, 33:690, 33:691, 50:1467, 50:1477, 50:1478, 50:1479,  
 33:692, 33:693, 33:694, 33:695, 50:1480, 50:1481, 50:1482, 50:1483,  
 33:696, 33:697, 33:698, 33:699, 01:310, 50:1554, 50:1557, 50:1573,  
 33:700, 33:701, 33:702, 33:703, 50:1593, 50:1600, 01:311, 06:504,  
 33:704, 33:705, 33:707, 33:708, 06:505, 06:506, 06:507, 06:508,  
 33:710, 33:711, 33:712, 33:713, 06:509, 06:540, 06:541, 06:542,  
 33:714, 33:715, 33:716, 33:717, 52:18, 52:19, 52:20, 52:21, 52:147,  
 33:718, 33:719, 33:721, 33:722, 52:189, 52:190, 52:197, 52:198,  
 33:723, 33:724, 33:725, 33:726, 52:199, 06:560, 06:561, 52:320,  
 33:727, 33:728, 33:729, 33:731, 52:341, 52:355, 52:359, 52:360,  
 33:732, 33:733, 33:734, 33:735, 52:492, 52:498, 52:499, 52:500,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

53:438, 53:439, 53:440, 53:441,  
 53:442, 53:444, 53:445, 53:446,  
 53:453, 53:454, 53:456, 53:458,  
 53:459, 53:460, 53:463, 53:486,  
 54:36, 06:668, 06:669, 06:670,  
 06:671, 06:672, 06:673, 06:675,  
 06:676, 06:677, 54:370, 54:371,  
 06:732, 06:747, 06:754, 54:1928,  
 54:1929, 54:1930, 54:1931, 06:828,  
 06:829, 06:830, 06:839, 57:10, 57:18,  
 57:39, 57:54, 57:73, 57:82, 57:89,  
 57:98, 57:108, 57:160, 57:161,  
 57:270, 57:283, 57:296, 57:316,  
 57:341, 57:351, 57:352, 57:353,  
 57:382, 57:384, 57:423, 57:424,  
 57:443, 57:444, 57:445, 57:446,  
 57:447, 57:448, 57:449, 57:450,  
 57:451, 57:468, 57:484, 57:485,  
 57:486, 57:529, 57:530, 57:658,  
 57:693, 57:694, 57:695, 57:696,  
 57:697, 02:65, 02:81, 02:105, 02:106,  
 02:121, 02:122, 02:127, 02:136,  
 02:149, 02:184, 02:189, 02:222,  
 02:223, 02:246, 02:256, 01:52, 01:53,  
 02:291, 02:353, 02:363, 02:365,  
 02:475, 02:476, 02:486, 02:487, 1247  
`\undefinedfonterror` ..... 1210  
`\unexpandable@protect` .....  
 ..... 20:211, 41:264,  
 06:232, 06:267, 06:273, 06:278, 1219  
`\unknownoptionerror` .....  
 ..... 50:1110, 50:1170, 50:1183  
`\unknowversion` ..... 1199  
`\unprocessedoptions` ..... 50:625,  
 50:734, 50:981, 50:1020, 50:1021,  
 50:1081, 50:1085, 50:1185, 51:85, 1233  
`\unused` ..... 14:15, 14:32, 14:59,  
 06:10, 06:17, 50:1578, 02:307, 1257  
`\unusedoptionlist` .....  
 ..... 20:18, 20:20, 20:88, 20:90, 20:145,  
 20:147, 50:12, 50:518, 50:519,  
 50:529, 50:530, 50:637, 50:645,  
 51:127, 51:128, 51:134, 51:161, 1032  
`\upline` ..... 42:297, 42:298, 42:304  
`\upordown` .....  
 ..... 42:195, 42:196, 42:204, 42:225, 42:273  
`\upvector` ..... 42:268, 42:304  
`\use@option` ..... 50:549, 50:565, 50:583,  
 50:599, 50:601, 50:618, 50:620, 50:630  
`\use@text@encoding` .....  
 ..... 21:150, 33:16, 33:1138  
`\vbsphack` ..... 18:220  
`\verb` ..... 37:725, 37:734, 37:737  
`\verbatim` 37:523, 37:569, 37:584, 37:593  
`\verb+visible+spacebox` ..... 37:583,  
 37:598, 37:610, 37:611, 37:624, 37:625  
`\verreq` ..... 30:464, 30:465  
`\viiipt` ..... 24:742  
`\viipt` ..... 24:741  
`\vipt` ..... 24:740  
`\vline` ..... 42:166, 42:178, 42:297  
`\vobeyspaces` ..... 37:499, 37:569,  
 37:586, 37:662, 37:676, 37:690, 37:737  
`\vobeytabs` ..... 37:499  
`\vppt` ..... 24:739  
`\vspace` ..... 18:377  
`\vspace@calcify` .....  
 ..... 18:80, 18:102, 18:242, 18:257,  
 18:384, 18:389, 18:399, 18:407,  
 37:435, 38:404, 41:62, 41:224, 42:148  
`\vspacer` ..... 18:377  
`\vtryfc` ..... 54:851, 54:859  
`\vvector` ..... 42:243, 42:258, 42:268  
`\warning` ..... 14:215  
`\ckptelt` ..... 20:440, 20:443  
`\whiledim` ..... 13:7, 42:123, 42:203  
`\whilenoop` ..... 13:3, 1227  
`\whilenum` ..... 13:3,  
 41:244, 42:104, 42:116, 42:336,  
 42:338, 42:361, 42:364, 42:393,  
 42:395, 42:416, 42:421, 42:730, 42:786  
`\whilesw` ..... 13:10, 54:266, 54:396,  
 54:405, 54:443, 54:453, 54:2275, 54:2315  
`\whileswnoop` ..... 13:10, 1227  
`\wholewidth` .....  
 ..... 40:199, 40:201, 40:202, 40:204,  
 40:206, 40:207, 40:208, 40:209, 42:2,  
 42:126, 42:129, 42:131, 42:299,  
 42:302, 42:350, 42:359, 42:406,  
 42:413, 42:564, 42:577, 42:589,  
 42:599, 42:678, 42:679, 42:727, 42:783  
`\width` ..... 18:528,  
 21:289, 21:294, 26:192, 30:625,  
 06:26, 40:204, 40:206, 40:282,  
 40:289, 40:550, 40:603, 40:610,  
 41:188, 41:219, 41:347, 41:366,  
 42:227, 42:299, 42:302, 42:325,  
 42:333, 42:350, 42:359, 42:384,  
 42:392, 42:406, 42:413, 42:564,  
 42:577, 42:727, 42:783, 45:395,  
 54:1855, 54:2264, 54:2298, 02:505, 1223  
`\wrglossary` ..... 46:25, 46:30, 1219  
`\wrindex` ..... 46:8, 46:13, 1219  
`\writeckpt` .....  
 ..... 20:330, 20:386, 20:425, 20:437  
`\writefile` ..... 20:32,  
 20:102, 20:160, 37:215, 44:183, 766  
`\writesetup` ..... 54:594

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\@wrong@font@char . . . . .  
     21:161, 24:562, 24:596, 24:609, 1230  
 \@wtryfc . . . . . 54:861, 54:871  
 \@x@protect . . . . . 06:105, 06:254  
 \@x@sf . . . . . 45:531, 45:533  
 \@xDeclareMathDelimiter . . . . .  
     . . . . . 28:984, 28:1040  
 \@xaddvskip . . . . .  
     . . . . . 18:220, 18:245, 18:260, 18:277  
 \@xarg . . . . . 42:163,  
     42:166, 42:175, 42:178, 42:185,  
     42:189, 42:190, 42:226, 42:228,  
     42:238, 42:239, 42:243, 42:253,  
     42:254, 42:258, 42:266, 42:274, 42:663  
 \@xargarraycr . . . . . 41:205, 41:214, 41:218  
 \@xargdef . . . . . 06:80  
 \@xarraycr . . . . . 41:202, 41:203  
 \@xbitor . . . . . 54:15, 54:17  
 \@xcentercr . . . . . 37:418, 37:425, 37:429  
 \@xdblarg . . . . . 06:798  
 \@xdblfloor . . . . . 45:264, 1218  
 \@xdim . . . . . 42:84, 42:93, 42:105, 42:107,  
     42:117, 42:119, 42:667, 42:731,  
     42:732, 42:733, 42:734, 42:740,  
     42:787, 42:788, 42:789, 42:790, 42:796  
 \@xeqnrcr . . . . . 38:388  
 \@exnoop . . . . . 41:238, 41:248  
 \@expast . . . . . 41:239, 41:240  
 \@xfloat 45:28, 45:29, 45:34, 45:266, 1219  
 \@xfootnote . . . . . 45:452, 45:455, 1196  
 \@xfootnotemark . . . . . 45:519, 45:523, 1196  
 \@xfootnotenext . . . . . 45:536, 45:539  
 \@xfootnotetext . . . . . 1196  
 \@xhline . . . . . 41:360, 41:361  
 \@xifnch . . . . . 06:786, 06:796  
 \@xipt . . . . . 24:746, 30:171, 30:173, 30:174  
 \@xipt . . . . . 24:745, 30:170  
 \@xivpt . . . . . 24:747, 30:172, 30:174  
 \@xmpar . . . . . 45:324, 45:325  
 \@xnewline . . . . .  
     . . . . . 18:61, 18:62, 18:73, 18:74, 18:94  
 \@xnext . . . . . 54:10, 54:11  
 \@xnthm . . . . . 43:5, 43:6  
 \@xobeysp . . . . . 18:467, 18:492,  
     37:504, 37:514, 37:518, 37:607,  
     37:611, 37:621, 37:625, 37:638, 438  
 \@xobeytab . . . . . 18:488, 37:506, 37:638  
 \@xprocess@ptions . . . . . 50:540,  
     50:557, 50:559, 50:575, 50:577, 50:591  
 \@xpt . . . . . 24:744, 30:169, 30:172, 30:173  
 \@xsect . . . . . 44:86, 44:87, 44:123  
 \@xtabcr . . . . . 41:56, 41:57  
 \@xtabularcr . . . . . 41:209, 41:210  
 \@xthm . . . . . 43:32, 43:38, 43:41  
 \@xtryfc . . . . . 54:848, 54:876  
 \@xtypein . . . . . 06:33, 06:35, 06:42  
 \@xverbatim . . . . . 37:518, 37:569  
 \@xviipt . . . . . 24:748, 30:173, 30:175  
 \@xxDeclarerMathDelimiter . . . . .  
     . . . . . 28:969, 28:973  
 \@xxpt . . . . . 24:749, 30:174, 30:175  
 \@xxvpt . . . . . 24:750, 30:175  
 \@xxxii . . . . . 12:2, 21:425,  
     21:427, 45:89, 45:158, 54:883,  
     54:884, 54:903, 54:904, 54:939,  
     54:940, 54:962, 54:963, 54:2041, 1230  
 \@xympar . . . . . 45:328, 45:332, 45:378  
 \@yarg . . . . . 42:163, 42:167,  
     42:175, 42:179, 42:185, 42:186,  
     42:195, 42:238, 42:244, 42:253,  
     42:259, 42:268, 42:270, 42:297, 42:663  
 \@yargarraycr . . . . . 41:206, 41:216, 41:220  
 \@yargd@f . . . . . 06:107  
 \@yargdef . . . . .  
     . . . . . 06:84, 06:94, 06:107, 06:123, 1232  
 \@ydim . . . . . 42:85, 42:94, 42:105, 42:108,  
     42:117, 42:119, 42:667, 42:735,  
     42:736, 42:737, 42:738, 42:739,  
     42:791, 42:792, 42:793, 42:794, 42:795  
 \@yeqnrcr . . . . . 38:388  
 \@ympar . . . . . 45:324, 45:329  
 \@ynthm . . . . . 43:5, 43:14  
 \@ythm . . . . . 43:32, 43:38, 43:41  
 \@ytryfc . . . . . 54:894, 54:913, 54:917  
 \@yyarg . . . . . 42:185,  
     42:186, 42:187, 42:190, 42:274, 42:663  
 \@ztryfc . . . . . 54:922, 54:933  
 \@zend . . . . . 220  
 \accent@spacefactor . . . . .  
     . . . . . 21:70, 21:73, 21:74, 1238  
 \active@math@prime . . . . .  
     . . . . . 38:253, 38:254, 54:592, 1238  
 \add@accent . . . . . 21:65, 21:67, 1231  
 \add@percent@to@temptokena . . . . .  
     . . . . . 37:202, 37:218, 37:220, 37:229, 767  
 \add@unicode@accent . . . . . 21:1039, 21:1053  
 \addto@hook . . . . .  
     . . . . . 24:152, 24:154, 24:738, 28:438,  
     28:574, 28:578, 28:595, 28:719,  
     28:725, 28:733, 28:749, 28:752,  
     28:755, 28:1188, 28:1195, 28:1198, 1194  
 \AddToHook . . . . . 303  
 \AddToHookNext . . . . . 303  
 \AddToHookNextWithArguments . . . . . 303  
 \AddToHookWithArguments . . . . . 303  
 \alloc@ . . . . .  
     . . . . . 04:22, 04:26, 04:38, 24:14, 02:90,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacel.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

|                                                                                                                                 |                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| 02:91, 02:92, 02:93, 02:94, 02:95,<br>02:96, 02:97, 02:98, 02:99, <u>02:226</u> , 1253                                          | \bx@GG . . . . . 54:41                                                                          |
| \alpha@elt . . . . . 28:45,<br>28:442, 28:669, 28:771, 28:1187, 28:1188                                                         | \bx@H . . . . . 54:31, 54:58                                                                    |
| \alpha@list . . . . . 28:41, <u>28:43</u> , 28:451,<br>28:657, 28:669, 28:714, 28:769,<br>28:770, 28:1183, 28:1189, 28:1190     | \bx@HH . . . . . 54:41                                                                          |
| \apptocmd . . . . . 304                                                                                                         | \bx@I . . . . . 54:31, 54:58                                                                    |
| \atveryend@DEPRECATED 52:580, 52:582                                                                                            | \bx@II . . . . . 54:41                                                                          |
| \best@size 26:438, 26:462, 26:468, 26:474                                                                                       | \bx@J . . . . . 54:31, 54:58                                                                    |
| \bf@def@ult . . . . . 29:397                                                                                                    | \bx@JJ . . . . . 54:41                                                                          |
| \bfdef@ult . . . . . 29:200,<br>29:271, 29:272, 29:273, 29:314,<br>29:315, 29:316, 29:401, 29:442, <u>643</u>                   | \bx@K . . . . . 54:32, 54:59                                                                    |
| \bfdefault@previous 29:267, 29:270,<br>29:310, 29:313, 30:113, 30:123, <u>643</u>                                               | \bx@KK . . . . . 54:42                                                                          |
| \bfseries@. . . . . 638                                                                                                         | \bx@L . . . . . 54:32, 54:59                                                                    |
| \bfseries@previous . . . . . 1254                                                                                               | \bx@LL . . . . . 54:42                                                                          |
| \bfseries@rm . . . . . 29:105, <u>29:128</u> ,<br>29:207, 29:210, 29:231, 29:258,<br>29:271, 29:314, 29:319, 29:355, <u>638</u> | \bx@M . . . . . 54:32, 54:59                                                                    |
| \bfseries@rm@kernel . . . . .<br>. . . . . 29:108, 29:131, 29:207, 29:234, <u>641</u>                                           | \bx@MM . . . . . 54:42                                                                          |
| \bfseries@sf . . . . . 29:106, <u>29:128</u> ,<br>29:211, 29:214, 29:232, 29:259,<br>29:272, 29:315, 29:320, 29:356, <u>641</u> | \bx@NN . . . . . 54:42                                                                          |
| \bfseries@sf@kernel . . . . .<br>. . . . . 29:109, 29:132, 29:211, 29:235                                                       | \bx@O . . . . . 54:33, 54:60                                                                    |
| \bfseries@tt . . . . . 29:107, <u>29:128</u> ,<br>29:215, 29:218, 29:233, 29:260,<br>29:273, 29:316, 29:321, 29:357, <u>641</u> | \bx@OO . . . . . 54:43                                                                          |
| \bfseries@tt@kernel . . . . .<br>. . . . . 29:110, 29:133, 29:215, 29:236                                                       | \bx@P . . . . . 54:33, 54:60                                                                    |
| \bm@b . . . . . 40:37                                                                                                           | \bx@PP . . . . . 54:43                                                                          |
| \bm@c . . . . . <u>40:37</u> , 40:50, 40:251, 40:346                                                                            | \bx@Q . . . . . 54:33, 54:60                                                                    |
| \bm@l . . . . . 40:37                                                                                                           | \bx@QQ . . . . . 54:43                                                                          |
| \bm@r . . . . . 40:37                                                                                                           | \bx@R . . . . . 54:33, 54:60                                                                    |
| \bm@s . . . . . 40:37                                                                                                           | \bx@RR . . . . . 54:43                                                                          |
| \bm@t . . . . . 40:37                                                                                                           | \bx@S . . . . . 54:38                                                                           |
| \botmark . . . . . 948                                                                                                          | \bx@SS . . . . . 54:44                                                                          |
| \bx@ . . . . . 1144                                                                                                             | \bx@T . . . . . 54:38                                                                           |
| \bx@A . . . . . 54:30, 54:57                                                                                                    | \bx@TT . . . . . 54:44                                                                          |
| \bx@AA . . . . . 54:40                                                                                                          | \bx@U . . . . . 54:38                                                                           |
| \bx@B . . . . . 54:30, 54:57                                                                                                    | \bx@UU . . . . . 54:44                                                                          |
| \bx@BB . . . . . 54:40                                                                                                          | \bx@V . . . . . 54:38                                                                           |
| \bx@C . . . . . 54:30, 54:57                                                                                                    | \bx@VV . . . . . 54:44                                                                          |
| \bx@CC . . . . . 54:40                                                                                                          | \bx@W . . . . . 54:39                                                                           |
| \bx@D . . . . . 54:30, 54:57                                                                                                    | \bx@WW . . . . . 54:45                                                                          |
| \bx@DD . . . . . 54:40                                                                                                          | \bx@X . . . . . 54:39                                                                           |
| \bx@E . . . . . 54:30, 54:57                                                                                                    | \bx@XX . . . . . 54:45                                                                          |
| \bx@EE . . . . . 54:40                                                                                                          | \bx@Y . . . . . 54:39                                                                           |
| \bx@F . . . . . 54:31, 54:58                                                                                                    | \bx@YY . . . . . 54:45                                                                          |
| \bx@FF . . . . . 54:41                                                                                                          | \bx@Z . . . . . 54:39                                                                           |
| \bx@G . . . . . 54:31, 54:58                                                                                                    | \bx@ZZ . . . . . 54:25, 54:45, 54:55                                                            |
|                                                                                                                                 | \c@bottomnumber 45:269, 45:274, <u>54:2330</u>                                                  |
|                                                                                                                                 | \c@dbltopnumber . . . . .<br>. . . . . 45:268, 45:283, 45:297, <u>54:2337</u>                   |
|                                                                                                                                 | \c@enumi . . . . . 39:251                                                                       |
|                                                                                                                                 | \c@enumii . . . . . 39:251                                                                      |
|                                                                                                                                 | \c@enumiv . . . . . 39:251, <u>1196</u>                                                         |
|                                                                                                                                 | \c@equation . . . . . <u>38:349</u> , 38:383, 38:543                                            |
|                                                                                                                                 | \c@errorcontextlines . . . . . 14:212, <u>1200</u>                                              |
|                                                                                                                                 | \c@footnote . . . . . 44:12, 45:397, 45:525                                                     |
|                                                                                                                                 | \c@localmathalphabets . . . . .<br>. . . . . <u>28:137</u> , 28:150, 28:282, 28:315, <u>607</u> |
|                                                                                                                                 | \c@mpfootnote . . . . . 40:454, 45:399                                                          |
|                                                                                                                                 | \c@ncel . . . . . 30:468, 30:469                                                                |
|                                                                                                                                 | \c@page . . . . . 34:3, 34:6,<br>34:7, 48:403, 48:462, 54:140, 54:1822                          |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\c@secnumdepth . . . . .  
     . . . . . 44:56, 44:71, 44:81, 44:140  
 \c@tocdepth . . . . . 44:140, 44:205, 44:228  
 \c@topnumber . . . . . 45:267, 45:271, 54:2326  
 \c@totalnumber . . . . . 45:270, 45:276, 54:2333  
 \c@totalpages . . . . . 53:348, 53:441  
 \calculate@math@sizes . . . . . 24:645, 26:219  
 \catcodetable@atletter . . . . . 04:93, 04:246  
 \catcodetable@initex . . . . . 04:93, 04:243  
 \catcodetable@latex . . . . . 04:93, 04:245  
 \catcodetable@string . . . . . 04:93, 04:244  
 \cdp@elt . . . . . . . . . 24:96,  
     24:116, 24:127, 24:128, 24:149,  
     24:152, 24:154, 28:352, 28:468,  
     28:543, 28:617, 28:698, 57:472, 57:473  
 \cdp@list . . . . . . . . . 24:98, 24:114,  
     24:128, 24:156, 24:157, 28:370,  
     28:470, 28:545, 28:619, 28:700, 57:473  
 \cf@encoding . . . . . . . . .  
     . . . . . 21:34, 21:41, 21:44, 21:51,  
     21:154, 24:256, 24:266, 24:276, 24:326  
 \ch@ck . . . . . . . . . 50:1259,  
     50:1278, 50:1390, 50:1411, 50:1503,  
     02:206, 02:207, 02:208, 02:209,  
     02:210, 02:236, 02:248, 02:249,  
     02:250, 02:251, 02:279, 02:281,  
     02:293, 02:294, 02:295, 02:296, 02:302  
 \char@if@alph . . . . . . . . . 06:806  
 \chardef@text@cmd . . . . . . . . . 21:3  
 \check@command . . . . . . . . . 06:187, 06:189  
 \check@icl . . . . . . . . . 32:9,  
     32:44, 32:49, 32:55, 32:63, 32:70, 32:72  
 \check@icr . . . . . . . . . 32:9, 32:44,  
     32:50, 32:56, 32:64, 32:73, 32:78, 1233  
 \check@mathfonts . . . . . . . . . 19:5,  
     21:302, 21:328, 21:360, 21:1254,  
     24:359, 24:363, 24:370, 24:372,  
     26:250, 28:320, 28:403, 33:672, 613  
 \check@nocorr . . . . . . . . . 1216  
 \check@nocorr@ . . . . . . . . . 32:46, 1235  
 \check@range . . . . . . . . . 26:379, 26:380  
 \check@singl . . . . . . . . . 26:378, 26:400  
 \cl@ckpt . . . . . . . . . 20:440, 22:35  
 \cl@page . . . . . . . . . 34:4  
 \col@number . . . . . . . . .  
     . . . . . 54:97, 54:150, 54:210, 54:222, 1209  
 \color@begingroup . . . . . . . . .  
     . . . . . 24:668, 24:729, 38:104, 38:134,  
     40:29, 40:110, 40:215, 40:442,  
     40:488, 40:505, 40:522, 41:47, 41:51,  
     45:475, 45:493, 45:511, 54:495, 1206  
 \color@endbox . . . . . . . . .  
     . . . . . 40:110, 45:253, 45:348, 45:366,  
     54:226, 54:635, 54:645, 54:694, 54:704  
 \color@endgroup . . . . . . . . .  
     . . . . . 24:673, 24:735, 38:104,  
     38:134, 40:29, 40:119, 40:134,  
     40:173, 40:194, 40:217, 40:470,  
     40:492, 40:509, 40:525, 41:49,  
     45:479, 45:497, 45:514, 54:499, 1206  
 \color@hbox . . . . . . . . . 40:110,  
     54:632, 54:642, 54:691, 54:701, 1220  
 \color@setgroup . . . . . . . . .  
     . . . . . 40:110, 40:173, 40:192, 1206  
 \color@vbox . . . . . . . . . 40:110, 45:96, 45:165,  
     45:339, 45:357, 45:381, 54:217, 1221  
 \conditionally@traceoff . . . . .  
     . . . . . 14:291, 07:285, 07:297, 398  
 \conditionally@traceon . . . . . 14:291, 07:321  
 \contionally@traceon . . . . . 398  
 \copy@kernel@robust@command . . . . .  
     . . . . . 06:588, 06:675  
 \count@ . . . . . . . . . 03:14,  
     03:15, 03:16, 03:17, 03:18, 03:20,  
     01:163, 01:164, 01:165, 24:703,  
     24:709, 24:711, 01:170, 26:22,  
     26:302, 26:304, 26:326, 26:327,  
     28:435, 28:437, 28:441, 28:800,  
     28:801, 28:802, 28:848, 28:849,  
     28:850, 28:909, 28:910, 28:911,  
     28:957, 28:958, 28:959, 28:997,  
     28:998, 28:999, 28:1005, 28:1006,  
     28:1007, 28:1051, 28:1052, 28:1053,  
     28:1059, 28:1060, 28:1061, 28:1120,  
     28:1121, 28:1122, 28:1128, 28:1129,  
     28:1130, 32:115, 32:118, 42:729,  
     42:730, 42:731, 42:734, 42:735,  
     42:738, 42:742, 06:169, 42:785,  
     42:786, 42:787, 42:790, 42:791,  
     42:794, 06:173, 42:798, 06:287,  
     06:312, 50:1302, 50:1304, 50:1305,  
     50:1306, 50:1435, 50:1437, 50:1438,  
     50:1439, 50:1524, 50:1526, 50:1527,  
     50:1528, 57:239, 57:240, 57:247,  
     57:249, 57:534, 57:535, 57:542,  
     57:544, 02:41, 02:191, 02:192,  
     02:197, 02:199, 02:205, 02:206,  
     02:207, 02:208, 02:209, 02:210,  
     02:211, 01:50, 02:505, 02:506, 1168  
 \counterwithin@s . . . . . 22:161, 22:162, 22:180  
 \counterwithin@x . . . . . 22:161, 22:164, 22:181  
 \counterwithout@s . . . . . 22:150, 22:151, 22:177  
 \counterwithout@x . . . . . 22:150, 22:153, 22:178  
 \curr@fontshape . . . . .  
     . . . . . 21:180, 24:88, 24:388, 24:396,  
     24:400, 24:402, 24:544, 24:550,  
     24:553, 24:562, 24:569, 24:571,  
     24:579, 24:585, 24:588, 24:596,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

24:603, 24:605, 25:454, 26:92,  
 26:100, 26:142, 26:169, 26:477,  
 26:497, 26:529, 26:545, 26:560,  
 28:374, 28:379, 29:556, 29:565, 1194  
 $\backslash\text{curr@math@size}$  .....  
 24:377, 26:256, 26:262, 26:267, 26:284  
 $\backslash\text{declare@command@copy}$  ..... 1253  
 $\backslash\text{declare@commandcopy}$  ..... 06:476,  
 06:480, 06:485, 06:488, 06:507, 98  
 $\backslash\text{declare@commandcopy@do}$  .....  
 ..... 06:488, 06:534, 06:535  
 $\backslash\text{declare@commandcopy@let}$  ..... 06:495,  
 06:499, 06:509, 06:610, 06:611, 99  
 $\backslash\text{declare@environmentcopy}$  .....  
 ..... 06:517, 06:521, 06:526, 06:529  
 $\backslash\text{declare@file@substitution}$  .....  
 ..... 52:247, 52:567, 53:517, 1040  
 $\backslash\text{declare@robustcommand}$  ..... 06:233  
 $\backslash\text{DeclareEncodingSubset@aux}$  .....  
 ..... 33:41, 33:43, 33:59  
 $\backslash\text{DeclareFontEncoding@}$  .....  
 ..... 24:122, 24:124,  
 24:139, 57:383, 57:403, 57:469, 1247  
 $\backslash\text{DeclareFontEncoding@saved}$  .....  
 ..... 57:383, 57:403, 57:485  
 $\backslash\text{DeclareFontShape@}$  ..... 24:21, 24:22  
 $\backslash\text{DeclareRobustCommand}$  ..... 312  
 $\backslash\text{DeclareSymbolFont@m@dropped}$  .....  
 ..... 28:460, 28:465, 28:508, 28:509  
 $\backslash\text{DeclareSymbolFontAlphabet@}$  .....  
 ..... 28:1175, 28:1178  
 $\backslash\text{DeclareUnicodeAccent@}$  .....  
 ..... 21:1046, 21:1048, 21:1052  
 $\backslash\text{def}$  ..... 322  
 $\backslash\text{default@ds}$  ..... 50:505, 50:536,  
 50:600, 50:619, 50:1108, 50:1110, 1198  
 $\backslash\text{default@family}$  .....  
 ... 24:129, 24:161, 24:512, 24:526,  
 24:529, 24:554, 24:589, 57:474, 1193  
 $\backslash\text{default@M}$  ..... 24:136,  
 24:176, 24:179, 24:183, 57:481, 1198  
 $\backslash\text{default@mextra}$  ..... 27:10, 27:89  
 $\backslash\text{default@series}$  .....  
 ..... 24:129, 24:162, 24:513,  
 24:527, 24:530, 24:551, 24:586, 57:474  
 $\backslash\text{default@shape}$  .....  
 ..... 24:130, 24:163, 24:514,  
 24:528, 24:531, 24:549, 24:584, 57:475  
 $\backslash\text{default@OT}$  .....  
 .. 24:170, 24:173, 24:183, 24:272, 1198  
 $\backslash\text{define@mathalphabet}$  .....  
 ..... 27:18, 27:131, 1193  
 $\backslash\text{define@mathgroup}$  .. 27:19, 27:135, 1193  
 $\backslash\text{define@newfont}$  ..... 24:380, 24:389  
 $\backslash\text{delayed@f@adjustment}$  .....  
 ... 24:290, 25:395, 25:396, 25:397,  
 25:399, 25:400, 25:411, 25:616,  
 25:617, 25:619, 25:620, 26:122,  
 26:126, 26:134, 26:138, 29:662, 657  
 $\backslash\text{delayed@merge@font@series}$  25:396,  
 25:460, 25:475, 25:514, 26:133, 26:136  
 $\backslash\text{delayed@merge@font@shape}$  .....  
 25:617, 25:666, 25:721, 26:132, 26:135  
 $\backslash\text{development@branch@name}$  .....  
 ..... 03:11, 03:39,  
 03:53, 03:54, 03:55, 03:62, 03:63, 03:64  
 $\backslash\text{dimen@}$  ..... 14:28, 14:29, 18:396,  
 18:401, 18:430, 18:435, 21:429,  
 21:430, 21:432, 21:433, 21:796,  
 21:797, 24:214, 24:216, 24:222,  
 24:235, 24:238, 24:242, 24:648,  
 24:649, 24:650, 24:654, 26:451,  
 26:452, 26:453, 26:454, 26:458,  
 33:102, 33:104, 33:928, 33:930,  
 38:72, 38:73, 38:199, 38:200, 38:201,  
 38:202, 40:587, 40:590, 41:176,  
 41:177, 54:512, 54:514, 54:535,  
 54:537, 02:41, 02:502, 02:503,  
 02:539, 02:540, 02:542, 02:544, 1232  
 $\backslash\text{dimen@i}$  ..... 02:41  
 $\backslash\text{dimen@ii}$  ..... 24:218, 24:223, 02:41  
 $\backslash\text{disable@package@load}$  .....  
 ..... 52:477, 53:475, 1055  
 $\backslash\text{display}$  ..... 38:204, 38:208, 38:209  
 $\backslash\text{do@add@percent@to@temptokena}$  ...  
 ..... 37:206, 37:212, 37:230  
 $\backslash\text{do@emfont@update}$  29:557, 29:561, 29:581  
 $\backslash\text{do@noligs}$  ..... 37:739, 37:744  
 $\backslash\text{do@subst@correction}$  .....  
 ..... 24:84, 26:482, 26:537  
 $\backslash\text{document@default@language}$  20:51,  
 20:52, 20:121, 20:122, 54:601, 57:286  
 $\backslash\text{document@select@group}$  .....  
 ..... 28:139, 28:145, 28:387, 607  
 $\backslash\text{dont@add@percent@to@temptokena}$  ..  
 ..... 37:205, 37:207, 37:231  
 $\backslash\text{dorestore@version}$  ... 28:114, 28:119  
 $\backslash\text{ds@}$  ..... 50:538, 50:1112, 1198  
 $\backslash\text{dt@pfalse}$  ..... 38:205  
 $\backslash\text{dt@ptrue}$  ..... 38:204  
 $\backslash\text{e@alloc}$  ..... 04:15, 04:49,  
 04:79, 04:89, 04:178, 04:182, 04:186,  
 04:194, 04:202, 04:210, 57:12, 57:47,  
 02:51, 02:52, 02:53, 02:55, 02:56,  
 02:63, 02:64, 02:66, 02:68, 02:79,  
 02:82, 02:84, 02:138, 02:230, 1253  
 $\backslash\text{e@alloc@attribute@count}$  .....  
 04:66, 04:74, 04:75, 04:76, 04:80, 04:237

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\ea@alloc@bytecode@count ... 04:70,
 04:197, 04:198, 04:199, 04:203, 04:253
\ea@alloc@ccodetable@count
 04:67, 04:84, 04:85, 04:86, 04:90, 04:241
\ea@alloc@chardef 04:48,
 04:178, 04:194, 04:202, 04:210,
 57:12, 02:60, 02:102, 02:211, 02:212
\ea@alloc@intercharclass@top .. 57:35
\ea@alloc@luachunk@count ... 04:71,
 04:205, 04:206, 04:207, 04:211, 04:255
\ea@alloc@luafunction@count
 ... 04:68, 04:173, 04:174, 04:175,
 04:179, 04:183, 04:187, 04:247, 04:249
\ea@alloc@top 04:47,
 04:80, 04:179, 04:183, 04:187,
 04:195, 04:203, 04:211, 57:12,
 02:55, 02:63, 02:102, 02:188, 02:259
\ea@alloc@whatsit@count 04:69,
 04:189, 04:190, 04:191, 04:195, 04:251
\ea@ch@ck .. 04:51, 04:55, 02:142, 02:152
\ea@insert@top
 ... 02:257, 02:259, 02:276, 02:291
\ea@mathgroup@top
 ... 28:56, 28:149, 28:150,
 28:187, 28:217, 02:79, 02:124, 607
\em@currfont 29:556, 29:567, 654
\em@force
 ... 29:565, 29:570, 29:573, 29:583, 654
\emfontdeclare@clist
 ... 29:546, 29:548, 29:552,
 29:557, 29:562, 29:568, 29:579, 653
\empty@sfcnt 26:490,
 26:491, 26:492, 26:506, 26:511, 26:563
\ENC@cmd 1213
\enc@update 24:257, 24:259,
 24:275, 24:278, 26:147, 26:173, 1213
\end@dblfloat 45:205
\end@float
 ... 45:189, 45:227, 45:243, 45:383, 1241
\endlinechar 322
\enlargethispage 952
\err@rel@i . 27:12, 27:99, 27:132, 27:136
\error@fontshape .. 24:507, 24:522,
 24:547, 24:582, 26:107, 26:527, 28:373
\escapechar 315
\et@xmaxfam .. 04:22, 04:26, 04:30, 04:38
\et@xmaxregs 04:29, 04:31,
 04:32, 04:33, 04:34, 04:35, 04:36, 04:37
\every@math@size
 ... 24:78, 26:235, 26:247, 1217
\every@size 1223
\execute@size@function
 ... 26:362, 26:390, 26:404, 26:421
\expand@font@defaults 29:37,
 29:148, 29:256, 29:279, 29:309,
 29:330, 29:354, 29:365, 29:396,
 29:397, 29:436, 29:438, 29:470,
 29:472, 29:501, 33:26, 33:83, 221
\expand@font@defaults 29:420
\expandafter 1055
\external@font 26:84, 26:87, 26:98,
 26:102, 26:104, 26:391, 26:405,
 26:467, 26:501, 26:569, 26:571, 26:573
\extra@def 27:9, 27:84, 1193
\extract@alph@from@version
 ... 24:622, 24:628,
 28:162, 28:193, 28:223, 28:255, 1196
\extract@default@composite
 ... 21:1062, 21:1069
\extract@default@composite@a
 ... 21:1071, 21:1075
\extract@default@composite@b
 ... 21:1073, 21:1077
\extract@font 24:403, 26:81
\extract@fontinfo 26:358, 26:365
\extract@rangefontinfo
 ... 26:375, 26:382, 26:401, 26:434
\extract@sizefn 26:350, 26:372
\f@... 1248
\f@baselineskip
 ... 21:870, 21:1220, 24:317, 24:324,
 24:533, 26:121, 26:167, 26:182,
 26:186, 26:201, 26:215, 26:226, 26:240
\f@depth 45:291, 54:347
\f@encoding 21:178, 04:260,
 04:275, 04:283, 24:251, 24:270,
 24:273, 24:274, 24:276, 24:326,
 24:383, 24:388, 24:407, 24:409,
 24:411, 24:416, 24:418, 24:449,
 24:511, 24:543, 24:578, 25:439,
 25:443, 25:652, 25:656, 26:91,
 26:128, 26:307, 26:517, 28:358, 1196
\f@family 24:279,
 24:287, 24:299, 24:309, 24:320,
 24:384, 24:388, 24:407, 24:409,
 24:411, 24:416, 24:418, 24:450,
 24:529, 24:554, 24:589, 25:439,
 25:443, 25:652, 25:656, 26:91,
 26:128, 28:358, 28:389, 29:151,
 29:178, 29:179, 29:258, 29:259,
 29:260, 29:281, 29:282, 29:283,
 29:319, 29:320, 29:321, 29:339,
 29:340, 29:341, 29:355, 29:356,
 29:357, 29:366, 29:367, 29:368,
 29:504, 29:517, 29:659, 29:677,
 29:693, 33:23, 33:27, 33:29, 33:31,
 33:63, 33:66, 33:80, 33:84, 33:86,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

33:88, 33:93, 33:100, 33:894, 33:897,  
 33:911, 33:920, 33:926, 33:1141, 651  
`\f@linespread` .... 24:320, 26:120,  
 26:166, 26:183, 26:184, 26:187,  
 26:195, 26:198, 26:209, 26:212, 1213  
`\f@series` ..... 19:14,  
 24:279, 24:310, 24:321, 24:385,  
 24:388, 24:530, 24:551, 24:586,  
 25:400, 25:410, 25:419, 25:429,  
 25:463, 25:482, 25:483, 25:652,  
 25:656, 26:125, 26:128, 26:131,  
 28:390, 29:144, 29:160, 29:162,  
 29:166, 29:167, 29:168, 29:189,  
 29:195, 29:198, 29:200, 29:222,  
 29:520, 29:524, 29:660, 29:678,  
 29:694, 29:739, 33:7, 33:613, 560  
`\f@series@saved` .... 26:125, 26:131  
`\f@shape` .....  
 ... 24:279, 24:289, 24:301, 24:311,  
 24:322, 24:386, 24:388, 24:531,  
 24:549, 24:584, 25:439, 25:443,  
 25:620, 25:627, 25:633, 25:643,  
 25:650, 25:654, 25:657, 25:660,  
 25:669, 25:676, 25:678, 25:713,  
 26:124, 26:128, 26:130, 28:391,  
 29:661, 29:679, 29:695, 29:740, 576  
`\f@shape@saved` .... 26:124, 26:130  
`\f@size` 21:180, 21:869, 21:1219, 24:88,  
 24:317, 24:323, 24:387, 24:532,  
 24:571, 24:605, 24:647, 24:648,  
 24:651, 24:652, 26:121, 26:142,  
 26:167, 26:169, 26:180, 26:200,  
 26:215, 26:218, 26:221, 26:226,  
 26:233, 26:240, 26:252, 26:255,  
 26:261, 26:267, 26:284, 26:285,  
 26:288, 26:293, 26:359, 26:366,  
 26:385, 26:387, 26:402, 26:453,  
 26:455, 26:457, 26:473, 26:474,  
 26:479, 26:493, 26:505, 26:510,  
 26:522, 26:530, 26:535, 26:561,  
 26:575, 29:556, 29:565, 33:102, 33:928  
`\f@user@size` .....  
 ..... 26:473, 26:478, 26:522, 26:535  
`\f@warn@break` ..... 1215  
`\famdef@ult` ..... 29:444  
`\filec@ntents` .....  
 ..... 50:1209, 50:1212, 50:1215,  
 50:1226, 50:1251, 50:1344, 50:1347,  
 50:1350, 50:1361, 50:1382, 50:1475,  
 50:1497, 50:1585, 50:1808, 1225  
`\filec@ntents@checkdir` .....  
 ..... 50:1232, 50:1234, 50:1252,  
 50:1367, 50:1369, 50:1383, 50:1482  
`\filec@ntents@force` .....  
 ..... 50:1228, 50:1363, 50:1478  
`\filec@ntents@noheader` .....  
 ..... 50:1230, 50:1365, 50:1480  
`\filec@ntents@nosearch` .....  
 ..... 50:1231, 50:1366, 50:1481  
`\filec@ntents@nowarn` ..... 50:1236  
`\filec@ntents@opt` .....  
 ..... 50:1212, 50:1215, 50:1217,  
 50:1347, 50:1350, 50:1352, 50:1477  
`\filec@ntents@OPTION` ..... 1010  
`\filec@ntents@overwrite` .....  
 ..... 50:1229, 50:1364, 50:1479  
`\filec@ntents@warning` .....  
 ..... 50:1237, 50:1239, 50:1284  
`\filec@ntents@where` .....  
 ..... 50:1233, 50:1235, 50:1264,  
 50:1368, 50:1370, 50:1395, 50:1483  
`\filename@area` ... 20:642, 20:657,  
 20:667, 20:699, 20:700, 20:704,  
 20:729, 20:732, 20:757, 20:773,  
 20:775, 01:230, 01:236, 01:243,  
 01:249, 01:256, 01:262, 01:269, 50:941  
`\filename@base` ..... 20:642,  
 20:657, 20:667, 20:699, 20:701,  
 20:704, 20:729, 20:732, 20:764,  
 20:773, 01:279, 01:286, 01:299, 50:942  
`\filename@dot` ..... 01:297, 01:303  
`\filename@dots` . 01:281, 01:283, 01:288  
`\filename@ext` ... 20:643, 20:658,  
 20:668, 20:695, 20:696, 20:699,  
 20:702, 20:704, 20:725, 20:726,  
 20:729, 20:732, 20:765, 01:278,  
 01:285, 01:295, 01:297, 50:943, 1046  
`\filename@parse` .....  
 ... 01:94, 20:640, 20:655, 20:665,  
 20:694, 20:724, 20:762, 01:226, 50:940  
`\filename@path` .....  
 ... 01:231, 01:232, 01:237, 01:244,  
 01:245, 01:250, 01:257, 01:258, 01:263  
`\filename@simple` .....  
 ..... 01:234, 01:247, 01:260,  
 01:270, 01:274, 01:276, 01:291, 01:293  
`\finish@module@release` .. 03:89, 03:93  
`\finph@nt` ..... 38:104,  
 38:106, 38:110, 38:111, 38:119, 38:120  
`\finsm@sh` ... 38:134, 38:141, 38:147,  
 38:153, 38:154, 38:158, 38:159, 788  
`\fix@penalty` ..... 32:101  
`\fixed@sfcnt` ... 26:565, 26:566, 26:567  
`\fl@ShowFloat` 54:1908, 54:1914, 54:1929  
`\fl@trace` ..... 54:242, 54:269,  
 54:325, 54:353, 54:360, 54:381,  
 54:428, 54:476, 54:529, 54:544,  
 54:545, 54:546, 54:547, 54:558,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

54:559, 54:560, 54:561, 54:562,  
 54:572, 54:785, 54:804, 54:823,  
 54:841, 54:843, 54:982, 54:986,  
 54:998, 54:999, 54:1000, 54:1001,  
 54:1007, 54:1010, 54:1018, 54:1022,  
 54:1033, 54:1038, 54:1043, 54:1044,  
 54:1045, 54:1046, 54:1053, 54:1056,  
 54:1064, 54:1075, 54:1081, 54:1086,  
 54:1091, 54:1097, 54:1098, 54:1103,  
 54:1108, 54:1109, 54:1110, 54:1118,  
 54:1122, 54:1127, 54:1131, 54:1136,  
 54:1147, 54:1148, 54:1150, 54:1168,  
 54:1177, 54:1183, 54:1192, 54:1195,  
 54:1201, 54:1211, 54:1215, 54:1225,  
 54:1231, 54:1237, 54:1243, 54:1250,  
 54:1252, 54:1258, 54:1263, 54:1265,  
 54:1267, 54:1275, 54:1280, 54:1286,  
 54:1291, 54:1297, 54:1311, 54:1312,  
 54:1315, 54:1336, 54:1345, 54:1351,  
 54:1360, 54:1363, 54:1370, 54:1380,  
 54:1384, 54:1396, 54:1402, 54:1407,  
 54:1412, 54:1416, 54:1420, 54:1421,  
 54:1428, 54:1433, 54:1437, 54:1444,  
 54:1453, 54:1457, 54:1461, 54:1462,  
 54:1466, 54:1467, 54:1477, 54:1483,  
 54:1489, 54:1495, 54:1499, 54:1505,  
 54:1507, 54:1515, 54:1520, 54:1525,  
 54:1533, 54:1542, 54:1547, 54:1552,  
 54:1554, 54:1559, 54:1561, 54:1572,  
 54:1578, 54:1588, 54:1594, 54:1598,  
 54:1599, 54:1604, 54:1605, 54:1611,  
 54:1614, 54:1615, 54:1616, 54:1623,  
 54:1624, 54:1625, 54:1633, 54:1638,  
 54:1650, 54:1651, 54:1658, 54:1661,  
 54:1669, 54:1673, 54:1677, 54:1678,  
 54:1682, 54:1683, 54:1693, 54:1699,  
 54:1709, 54:1715, 54:1719, 54:1720,  
 54:1726, 54:1727, 54:1734, 54:1737,  
 54:1738, 54:1739, 54:1747, 54:1748,  
 54:1749, 54:1758, 54:1763, 54:1776,  
 54:1778, 54:1785, 54:1788, 54:1797,  
 54:1801, 54:1805, 54:1806, 54:1810,  
 54:1811, 54:1863, 54:1868, 54:1874,  
 54:1884, 54:1891, 54:1905, 54:1906,  
 54:1910, 54:1921, 54:1933, 54:2031,  
 54:2044, 54:2045, 54:2049, 54:2052,  
 54:2054, 54:2057, 54:2060, 54:2062,  
 54:2103, 54:2110, 54:2115, 54:2121,  
 54:2126, 54:2130, 54:2136, 54:2144,  
 54:2146, 54:2153, 54:2158, 54:2163,  
 54:2165, 54:2171, 54:2173, 54:2180,  
 54:2209, 54:2211, 54:2226, 54:2228,  
 54:2242, 54:2267, 54:2271, 54:2276,  
 54:2288, 54:2305, 54:2310, 54:2318  
  
 \f1@tracemessage . . . . .  
 .. 54:1905, 54:1922, 54:1931, 54:1933  
 \f1@traceval . . . . .  
 .. 54:1915, 54:1916, 54:1917,  
 54:1918, 54:1921, 54:1930, 54:1933  
 \float@count . . . . . 02:51,  
 02:52, 02:53, 02:62, 02:188, 02:205,  
 02:211, 02:213, 02:214, 02:222, 02:230  
 \fmtversion@topatch . . . . . 57:655,  
 57:657, 57:669, 57:670, 57:682, 57:690  
 \font@info 26:99, 26:365, 26:434, 26:439  
 \font@name .. 21:179, 21:182, 24:86,  
 24:194, 24:196, 24:379, 24:394,  
 24:570, 24:604, 26:84, 26:88, 26:90,  
 26:105, 26:141, 26:144, 26:154,  
 26:168, 26:171, 26:330, 26:331,  
 26:332, 26:333, 26:334, 26:339, 1193  
 \font@submax 26:441, 26:470, 26:471,  
 37:53, 37:55, 37:118, 37:120, 37:157,  
 37:159, 57:299, 57:301, 57:310, 1208  
 \fps@dbl . . . . . 45:34  
 \freeze@math@version . 28:154, 28:273  
 \frozen@everydisplay . . . . .  
 .. 24:349, 24:353, 24:368, 24:370, 611  
 \frozen@everymath . . . . .  
 .. 24:349, 24:361, 24:372, 611  
 \g@addto@macro . . . . .  
 .. 09:61, 25:731, 28:403, 50:133,  
 50:460, 50:1120, 50:1136, 50:1137,  
 53:352, 53:358, 53:408, 06:918, 229  
 \G@refundefined . . . . . 1230  
 \G@refundefinedfalse . . . . . 35:5, 739  
 \G@refundefinedtrue .. 35:3, 35:16,  
 35:33, 35:50, 36:203, 36:210, 36:220,  
 36:226, 47:41, 47:68, 47:85, 1230  
 \gen@sfcnt . . . . . 26:502, 26:503, 26:504  
 \genb@sfcnt 26:507, 26:508, 26:509, 1229  
 \genb@x . . . . . 26:510, 26:512  
 \genb@y . . . . . 26:512  
 \get@cdp . . . . . 28:571, 28:579, 28:612  
 \get@external@font 26:83, 26:96, 26:536  
 \getanddefine@fonts 24:617, 24:635,  
 26:320, 28:59, 28:87, 28:132, 28:159,  
 28:190, 28:220, 28:251, 28:294,  
 28:333, 28:438, 28:522, 28:576,  
 28:578, 28:595, 28:718, 28:719,  
 28:751, 28:752, 28:1194, 28:1195, 610  
 \glb@currsize . . . . .  
 .. 20:41, 20:111, 20:169, 24:346,  
 26:217, 26:252, 26:256, 26:262, 26:285  
 \glb@settings . . . . .  
 .. 24:347, 26:217, 26:264, 26:295  
 \gobble@finish@module@release . . . . .  
 .. 03:109, 03:111, 03:170

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\gobble@font@spec 1227
\group@elt 28:35, 28:436,
28:483, 28:484, 28:515, 28:519, 28:1226
\group@list 28:440, 28:490,
28:513, 28:518, 28:519, 28:568,
28:794, 28:842, 28:903, 28:988,
28:991, 28:1042, 28:1045, 28:1111,
28:1114, 28:1181, 28:1232, 1236
\h@false 38:81
\h@true 38:82, 38:83
\hb@xt@ 21:425, 38:210, 38:376,
38:455, 38:470, 38:482, 38:509,
38:540, 06:29, 40:48, 40:63, 40:86,
40:104, 40:248, 40:267, 40:614,
40:618, 40:619, 40:620, 41:37,
42:31, 42:43, 42:62, 42:74, 42:105,
42:117, 42:265, 42:299, 42:302,
42:305, 42:307, 42:314, 42:373,
42:452, 42:587, 42:597, 42:740,
42:796, 44:218, 44:241, 44:248,
54:634, 54:644, 54:693, 54:703,
54:1847, 54:2261, 54:2262, 54:2266,
54:2293, 54:2294, 54:2300, 02:549, 1224
\hexnumber@ . 28:815, 28:823, 28:858,
28:866, 28:887, 28:897, 28:923,
28:931, 28:939, 28:948, 28:951,
28:960, 28:961, 28:1000, 28:1008,
28:1054, 28:1062, 28:1081, 28:1082,
28:1092, 28:1093, 28:1098, 28:1124,
28:1132, 28:1137, 28:1139, 29:645
\hgl@ 02:504, 02:505
\hmode@bgroup 21:67, 21:75, 21:327,
21:356, 21:390, 21:396, 21:427,
21:438, 21:445, 21:476, 21:483,
21:486, 21:488, 21:496, 21:512,
21:730, 21:760, 21:766, 21:798,
21:805, 21:833, 21:836, 21:893,
21:1253, 32:7, 33:625, 33:632, 1238
\hmode@start@before@group
. 21:68, 21:151, 21:153, 21:159, 21:184
\hyper@nopatch@longtable 55:241
\if@afterindent 44:124, 44:131
\if@compatibility 50:2, 50:676
\if@endpe
37:338, 37:353, 37:370, 37:380, 39:140, 1269
\if@eqnsw
38:357, 38:423
\if@fcolmade 54:97, 54:266, 54:396,
54:405, 54:443, 54:453, 54:783,
54:803, 54:821, 54:850, 54:930,
54:2208, 54:2225, 54:2275, 54:2315
\if@files w
20:5, 20:36,
20:106, 20:164, 20:298, 20:310,
20:331, 20:360, 20:372, 20:387,
20:407, 20:419, 20:426, 20:438,
37:21, 37:59, 37:87, 37:124, 37:149,
37:163, 44:153, 47:4, 47:8, 47:39,
47:48, 47:56, 47:67, 47:84, 50:1262,
50:1279, 50:1393, 50:1412, 53:361, 1219
\if@firstamp 41:251
\if@firstcolumn
. 54:97, 54:248, 54:281, 54:398,
54:446, 54:1819, 54:2239, 54:2284, 949
\if@font@series@context
. 29:509, 29:528, 29:536, 651
\if@forced@series 25:401, 29:143, 1250
\if@ignore
28:301, 28:340,
37:4, 37:339, 37:356, 37:372, 37:381
\if@in@minipage@env ... 40:423, 40:445
\if@includeinrelease
. 03:73, 03:76, 03:132, 06:872
\if@inlabel
39:28, 39:65,
39:102, 39:184, 39:207, 54:163, 54:190
\if@inmath 1232
\if@insert
54:97, 54:1061,
54:1173, 54:1207, 54:1341, 54:1376,
54:1450, 54:1539, 54:1666, 54:1794
\if@listfiles@hashes
. 20:743, 20:778, 20:785, 20:789
\if@listfiles@sizes
. 20:743, 20:779, 20:783
\if@minipage
. 16:26, 18:240, 18:255, 18:272,
18:291, 18:324, 18:358, 37:528,
37:550, 39:173, 40:420, 41:79, 45:20
\if@mparswitch
54:97, 54:1821
\if@multiplelabels
35:93
\if@mypkg@draft 1027
\if@negarg 42:157, 42:198, 42:212, 42:273
\if@newlist
37:570,
39:29, 39:33, 39:69, 39:78, 39:106,
39:190, 54:603, 54:648, 54:661, 54:707
\if@nmbrlist
39:33, 39:225
\if@no@font@opt . 27:16, 27:110, 27:129
\if@nobreak
. 18:121, 18:148, 18:293, 18:326,
18:360, 20:203, 20:215, 39:191,
39:216, 40:383, 40:404, 44:47,
44:128, 45:180, 45:373, 48:241,
49:33, 49:44, 49:58, 49:66, 54:167,
54:194, 54:337, 54:1152, 54:1318, 1234
\if@noitemarg
39:32, 39:223
\if@noparitem
39:30, 39:181
\if@noparlist
39:31, 39:114
\if@noskipsec
18:148,
39:58, 40:384, 40:405, 44:38,
44:40, 44:97, 45:374, 54:157, 54:184
\if@ovb
42:427,
42:494, 42:525, 42:550, 42:561, 42:574

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltphyphen.dtx, 57=ltfinal.dtx

```

\if@ovhline 42:459, 42:589
\if@ovl 42:427,
 42:492, 42:521, 42:546, 42:591, 42:600
\if@ovr 42:427,
 42:491, 42:520, 42:545, 42:588, 42:598
\if@ovt 42:427,
 42:493, 42:524, 42:549, 42:566, 42:578
\if@ovvline 42:459, 42:564
\if@partsw . 20:5, 20:302, 20:364, 20:411
\if@pboxsw 40:351, 40:373, 40:529
\if@reversemargin 54:97, 54:1824
\if@rjfield 41:19, 41:33
\if@skipping@Module 03:137, 03:149, 03:152
\if@specialpage . 54:97, 54:610, 54:668
\if@tempswa .. 12:9, 20:308, 20:370,
 20:417, 21:1542, 24:99, 24:713,
 28:471, 28:546, 28:620, 28:701,
 28:1225, 37:61, 37:126, 37:165,
 37:535, 37:556, 47:95, 50:1291,
 50:1424, 50:1513, 52:292, 52:293,
 52:315, 52:316, 52:336, 52:337,
 54:994, 54:1030, 54:1630, 54:1755,
 02:270, 01:62, 01:63, 01:64, 1241
\if@test 54:12, 54:13, 54:891,
 54:910, 54:950, 54:972, 54:1036,
 54:1120, 54:1129, 54:1278, 54:1289,
 54:1431, 54:1518, 54:1636, 54:1761
\if@twocolumn 20:26,
 20:96, 20:153, 45:32, 45:210, 45:235,
 54:97, 54:141, 54:269, 54:280,
 54:397, 54:445, 54:469, 54:785,
 54:841, 54:1818, 54:2210, 54:2227
\if@twoside 54:97, 54:140, 54:613, 54:671
\ifdt@p 38:203, 38:205
\IfFileExists@ 20:471, 20:498
\IfFileExists@@ 20:498
\ifG@refundefined 35:3, 35:4, 35:5
\ifh@ 38:76, 38:114, 38:123
\ifin@ 21:1562, 21:1565, 27:50, 27:52,
 28:1, 28:22, 28:425, 28:567, 28:569,
 28:630, 28:643, 28:713, 28:715,
 28:743, 28:795, 28:809, 28:843,
 28:855, 28:904, 28:920, 28:989,
 28:992, 28:1013, 28:1043, 28:1046,
 28:1109, 28:1112, 28:1115, 28:1182,
 28:1184, 28:1213, 29:210, 29:214,
 29:218, 50:236, 50:254, 50:548, 50:582
\ifmath@fonts 24:204, 26:222
\ifmaybe@ic 32:82, 32:91, 1241
\ifnot@nil 26:343, 26:360, 26:381
\ifrestore@version 1232
\iftc@forced .. 33:868, 33:879, 33:1147
\ifv@ 38:75, 38:113, 38:122
\ifx 323
\in@ 21:1560,
 21:1563, 27:49, 27:51, 28:1, 28:21,
 28:424, 28:566, 28:568, 28:626,
 28:639, 28:712, 28:714, 28:741,
 28:793, 28:804, 28:841, 28:852,
 28:902, 28:916, 28:987, 28:990,
 28:1010, 28:1041, 28:1044, 28:1106,
 28:1110, 28:1113, 28:1180, 28:1183,
 28:1211, 29:208, 29:212, 29:216,
 50:235, 50:252, 50:545, 50:581, 564
\in@C 25:489, 25:490,
 25:492, 25:493, 28:5, 28:6, 28:7, 28:9
\in@false 28:10
\in@true 28:12
\init@restore@glb@settings 26:265, 26:268, 26:270
\init@restore@version 28:62,
 28:91, 28:108, 28:123, 28:124, 1232
\init@series@setup 25:732, 25:737, 29:120, 29:206, 29:246
\input@path 01:93,
 01:115, 01:117, 01:123, 01:125,
 01:131, 01:133, 20:484, 20:538,
 20:560, 20:587, 20:604, 01:138,
 01:140, 01:150, 01:217, 52:275, 1052
\insc@unt 54:61, 02:37,
 02:51, 02:52, 02:53, 02:62, 02:90,
 02:91, 02:92, 02:94, 02:247, 02:248,
 02:249, 02:250, 02:251, 02:252,
 02:263, 02:264, 02:265, 02:266,
 02:267, 02:271, 02:273, 02:292,
 02:293, 02:294, 02:295, 02:296, 02:297
\install@mathalphabet 24:612, 24:629, 24:636,
 28:444, 28:447, 28:573, 28:574,
 28:671, 28:723, 28:726, 28:733,
 28:748, 28:749, 28:756, 28:1196, 28:1198
\is@orange 26:376, 26:377
\kernel@ifnextchar .. 03:90, 06:81,
 06:100, 06:150, 50:440, 06:783, 06:798
\kernel@make@fragile 18:25, 18:26, 18:27, 18:28, 18:29,
 21:172, 21:173, 37:487, 37:488,
 37:489, 38:90, 38:91, 38:92, 38:93,
 38:179, 38:180, 38:181, 41:160,
 41:161, 41:162, 42:822, 42:823,
 42:824, 42:825, 42:826, 42:827,
 42:828, 42:829, 42:830, 42:831,
 42:832, 42:833, 44:23, 44:24, 44:25,
 44:26, 44:27, 49:72, 49:73, 06:387,
 06:899, 06:900, 06:901, 06:902,
 06:903, 06:904, 06:905, 06:906,
 06:907, 06:908, 06:909, 06:910,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

06:911, 06:912, 06:913, 06:914, 29  
 $\backslash 1@ngrel@x$  ..... 06:74, 06:75, 06:76, 06:120, 06:167  
 $\backslash 1@nohyphenation$  ..... 37:532, 37:724, 57:283, 1246  
 $\backslash last@fontshape$  ..... 24:545, 24:563, 24:580, 24:597  
 $\backslash latexrelease@postltcmd$  ..... 07:3014  
 $\backslash latexrelease@postltexpl$  ..... 05:132  
 $\backslash leavevmode@ifvmode$  ..... 18:537, 18:538, 18:546,  
 30:631, 30:633, 30:635, 30:637,  
 38:115, 38:154, 38:219, 38:239, 38:240  
 $\backslash load@onefile@withoptions$  ..... 50:945, 50:987, 50:1099, 1002  
 $\backslash load@onefilewithoptions$  ..... 50:892, 50:1049, 50:1600, 1001  
 $\backslash lower@bound$  ... 26:386, 26:387, 26:398  
 $\backslash lst@vskip$  ..... 414  
 $\backslash LT@cols$  ..... 1168  
 $\backslash ltx@sh@ft$  ... 21:390, 21:397, 21:476,  
 21:484, 21:760, 21:767, 02:541, 1241  
 $\backslash m@ne$  ..... 02:39, 382  
 $\backslash m@th$  ..... 19:13, 30:344,  
 30:466, 30:468, 30:469, 30:472,  
 30:513, 30:537, 30:540, 30:544,  
 30:547, 30:554, 30:557, 30:564,  
 30:567, 30:649, 38:68, 38:71, 38:106,  
 38:140, 38:147, 38:167, 38:169,  
 38:185, 38:204, 38:367, 38:470,  
 38:482, 38:509, 38:520, 40:351,  
 40:373, 40:555, 41:181, 44:214,  
 44:237, 45:400, 45:409, 45:416,  
 45:437, 45:444, 02:521, 02:533, 1230  
 $\backslash makeph@nt$  ..... 38:101, 38:103  
 $\backslash makesm@sh$  ..... 38:131, 38:133  
 $\backslash mandatory@arg$  ..... 26:414,  
 26:501, 26:505, 26:510, 26:517,  
 26:519, 26:524, 26:526, 26:531,  
 26:533, 26:546, 26:562, 26:569, 26:571  
 $\backslash math@bgroup$  ..... 24:643,  
 26:306, 26:312, 28:53, 28:81, 28:146,  
 28:175, 28:184, 28:206, 28:214,  
 28:245, 32:130, 32:131, 32:138, 608  
 $\backslash math@egroup$  ..... 24:643, 26:310,  
 26:311, 32:131, 32:132, 32:139, 1194  
 $\backslash math@famname$  ..... 1193  
 $\backslash math@fonts$  ..... 24:613,  
 24:618, 26:232, 26:336, 28:60,  
 28:89, 28:160, 28:191, 28:221, 28:253  
 $\backslash math@fontsfalse$  ..... 19:7, 21:302, 21:329,  
 21:360, 21:1255, 24:77, 24:206,  
 24:216, 24:239, 33:103, 33:673, 33:929  
 $\backslash math@fontstrue$  ..... 24:204, 24:655  
 $\backslash math@group$  ..... 1193  
 $\backslash math@version$  ..... 24:7,  
 24:336, 24:617, 24:621, 24:623,  
 24:624, 24:626, 26:230, 28:56, 28:59,  
 28:64, 28:65, 28:69, 28:84, 28:88,  
 28:93, 28:94, 28:98, 28:111, 28:112,  
 28:113, 28:126, 28:127, 28:128,  
 28:149, 28:151, 28:153, 28:154,  
 28:159, 28:163, 28:165, 28:167,  
 28:171, 28:187, 28:190, 28:194,  
 28:196, 28:198, 28:202, 28:217,  
 28:220, 28:224, 28:226, 28:228,  
 28:232, 28:248, 28:252, 28:256,  
 28:258, 28:260, 28:264, 29:617, 607  
 $\backslash math@version$  ..... 1193  
 $\backslash mathchar@type$  ..... 28:887,  
 28:897, 28:948, 28:951, 28:960,  
 28:976, 28:1081, 28:1092, 28:1155  
 $\backslash mathph@nt$  ..... 38:99, 38:105  
 $\backslash mathsm@sh$  ..... 38:129, 38:138, 38:139, 38:145, 38:146  
 $\backslash maybe@ic$  ..... 32:63, 32:64, 32:83  
 $\backslash maybe@ic@$  ..... 32:83  
 $\backslash maybe@icfalse$  ..... 32:97  
 $\backslash maybe@icttrue$  ..... 32:87  
 $\backslash maybe@load@fontshape$  ..... 21:71,  
 25:476, 25:515, 26:127, 29:164, 473  
 $\backslash maybe@update@bfseries@defaults$  ..  
 ..... 29:38, 29:257, 29:266, 29:304  
 $\backslash maybe@update@mdseries@defaults$  ..  
 ..... 29:39, 29:280, 29:289, 29:305  
 $\backslash mb@b$  ..... 40:76, 40:87, 40:95, 40:105  
 $\backslash mb@l$  ..... 40:76, 40:80, 40:86,  
 40:95, 40:99, 40:104, 42:137, 42:141  
 $\backslash mb@r$  ..... 40:76, 40:80, 40:86,  
 40:95, 40:99, 40:104, 42:137, 42:141  
 $\backslash mb@t$  ..... 40:77, 40:84, 40:96, 40:103  
 $\backslash md@def@ult$  ..... 29:397  
 $\backslash mddef@ult$  ..... 29:198,  
 29:293, 29:294, 29:295, 29:334,  
 29:335, 29:336, 29:402, 29:443, 1251  
 $\backslash mddefault@previous$  29:290, 29:292,  
 29:331, 29:333, 30:114, 30:124, 1254  
 $\backslash mdseries@..$  ..... 636  
 $\backslash mdseries@previous$  ..... 1254  
 $\backslash mdseries@rm$  ..... 29:111,  
 29:127, 29:128, 29:229, 29:237,  
 29:281, 29:293, 29:334, 29:339, 29:366  
 $\backslash mdseries@sf$  29:112, 29:128, 29:238,  
 29:282, 29:294, 29:335, 29:340, 29:367  
 $\backslash mdseries@tt$  29:113, 29:128, 29:239,  
 29:283, 29:295, 29:336, 29:341, 29:368  
 $\backslash meaning$  ..... 324

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphyphen.dtx, 57=ltfinal.dtx

```

\merge@font@series 25:408, 25:425,
 25:426, 25:507, 25:509, 26:133, 560
\merge@font@series@ . . .
 25:428, 25:433, 25:510
\merge@font@series@without@substitution
 25:460, 25:475, 25:512, 26:136, 562
\merge@font@series@without@substitution@
 25:460, 25:513
\merge@font@shape 25:626,
 25:639, 25:640, 25:711, 25:716, 26:132
\merge@font@shape@
 25:642, 25:647, 25:717
\merge@font@shape@without@substitution
 25:666, 25:719, 26:135
\merge@font@shape@without@substitution@
 25:666, 25:720
\mv@{version} 610
\mv@{version}@frozen 607
\mv@{version}@reset 610
\n@space
 30:632, 30:634, 30:636, 30:638,
 30:643, 30:644, 30:645, 30:646, 30:649
\new@command 06:78,
 06:77, 06:131, 06:165, 06:184, 06:251
\new@environment 06:147, 06:146, 06:159
\new@fontshape 27:2, 27:4, 27:22, 27:24
\new@label@record
 36:57, 36:128, 36:278, 37:25, 749
\new@mathalphabet 28:624, 28:645, 28:656
\new@mathgroup 04:27, 24:14, 28:474,
 02:78, 02:80, 02:98, 02:100, 1226
\new@mathversion
 28:20, 28:407, 28:415, 28:420, 28:423
\new@module@skip 03:138, 03:150, 03:152
\new@moduledate
 03:82, 03:101, 03:104, 03:152
\new@modulename 03:96, 03:152
\new@symbolfont 28:475, 28:517
\newcommand 305
\NewCommandCopy 312
\NewDocumentCommand 305
\nnewlinechar 322
\newmathalphabet@ 27:14
\newmathalphabet@@ 27:109
\newmathalphabet@CC 27:15, 27:109
\nfss@catcodes 24:19, 24:120,
 24:412, 24:413, 24:420, 24:467,
 30:40, 30:45, 30:53, 30:142, 54:3, 1229
\nfss@text 21:315, 21:317, 29:648,
 32:5, 32:122, 35:17, 35:34, 35:51, 1216
\no@alphabet@error 24:4,
 28:443, 28:445, 28:661, 28:662,
 28:676, 28:685, 28:771, 28:772, 1208
\no@alphabet@help 1194
\no@version@warning 1194
\noaccents@ 24:658, 30:136
\noexpand 324
\non@alpherr 24:637,
 24:639, 28:72, 28:101, 28:117,
 28:174, 28:205, 28:235, 28:267, 28:1233
\not@base 29:720, 29:724, 29:725,
 29:726, 29:727, 29:728, 29:729,
 29:730, 29:731, 29:732, 29:733, 29:734
\not@math@alphabet
 25:527, 25:532, 25:537, 25:683,
 25:687, 25:690, 25:693, 25:696,
 25:699, 25:702, 25:705, 29:5, 29:8,
 29:11, 29:14, 29:17, 29:20, 29:23,
 29:26, 29:29, 29:255, 29:278, 29:308,
 29:329, 29:353, 29:364, 29:381,
 29:384, 29:406, 29:411, 29:416,
 29:449, 29:454, 29:459, 29:475,
 29:478, 29:481, 29:484, 29:487, 29:597
\now@and@everyjob
 04:215, 04:221, 21:1011
\no@align 21:390, 21:397,
 21:476, 21:484, 21:760, 21:767, 02:535
\non@line 14:8, 14:15, 14:214, 29:144,
 29:147, 37:252, 37:273, 37:291,
 37:301, 40:189, 50:953, 50:1088,
 08:534, 08:607, 08:1478, 08:1513, 407
\operator@font 30:650, 38:3, 38:4,
 38:5, 38:6, 38:7, 38:8, 38:9, 38:10,
 38:11, 38:12, 38:13, 38:14, 38:15,
 38:16, 38:17, 38:18, 38:19, 38:20,
 38:21, 38:22, 38:23, 38:24, 38:25,
 38:26, 38:27, 38:28, 38:29, 38:30,
 38:31, 38:32, 38:33, 38:34, 38:37, 38:40
\optional@arg
 26:415, 26:494, 26:496, 26:568, 26:571
\nouter@nobreak 45:181,
 45:251, 45:255, 45:346, 45:364, 1221
\np@ 02:311
\np@. 1249
\np@enum 816
\np@equation 38:364, 38:518
\np@renwd 1204
\np@selectfont 26:119, 1194
\npar 169
\npar@deathcycles
 39:56, 39:77, 39:79, 39:80
\npatch@level 03:1,
 03:39, 03:44, 03:46, 03:48, 03:52,
 03:61, 57:658, 57:670, 57:672, 1244
\npatchcmd 305
\nph@nt 38:81, 38:82, 38:83, 38:97
\npickup@font 21:181,
 24:195, 24:378, 24:572, 24:606,

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

26:143, 26:170, 26:331, 26:333, 26:335
\picture 42:21
\pkgcls@arg 50:1611, 50:1742
\pkgcls@candidate
..... 50:1598, 50:1620, 50:1696,
50:1700, 50:1704, 50:1773, 50:1776
\pkgcls@debug ... 50:1588, 50:1604,
50:1605, 50:1606, 50:1607, 50:1608,
50:1672, 50:1673, 50:1674, 50:1675,
50:1684, 50:1689, 50:1707, 50:1716,
50:1732, 50:1766, 50:1767, 50:1768
\pkgcls@ext ... 50:1612, 50:1660, 1268
\pkgcls@innerdate 50:1593,
50:1645, 50:1648, 50:1654, 50:1794
\pkgcls@mindate 50:1625,
50:1634, 50:1650, 50:1655, 1020
\pkgcls@name 50:1610, 50:1660
\pkgcls@parse@date@arg
..... 50:1619, 50:1630
\pkgcls@parse@date@arg@
..... 50:1636, 50:1639
\pkgcls@parse@date@arg@version ..
..... 50:1646, 50:1667
\pkgcls@releasedate
.. 50:1598, 50:1701, 50:1705, 50:1777
\pkgcls@rollbackdate@error
..... 50:1697, 50:1756, 50:1774
\pkgcls@show@selection
.. 50:1724, 50:1730, 50:1780, 50:1785
\pkgcls@targetdate
..... 50:1593, 50:1632, 50:1640,
50:1643, 50:1644, 50:1648, 50:1656,
50:1657, 50:1670, 50:1678, 50:1693,
50:1695, 50:1725, 50:1737, 50:1739,
50:1764, 50:1770, 50:1772, 1020
\pkgcls@targetlabel
..... 50:1593, 50:1633,
50:1653, 50:1668, 50:1680, 50:1712,
50:1746, 50:1784, 50:1787, 1020
\pkgcls@use@this@release
..... 50:1681, 50:1698,
50:1700, 50:1713, 50:1723, 50:1776
\pr@Cs 38:259, 38:267
\pr@@t 38:262, 38:268
\pr@mCs 38:256, 38:257
\preload@sizes ... 27:11, 27:94, 1193
\prepare@family@series@update ...
..... 29:117,
29:141, 29:243, 29:407, 29:412,
29:417, 29:450, 29:455, 29:460, 648
\pretocmd 304
\prim@s 38:253, 38:255, 38:267
\prime@s 38:254
\process@table
..... 20:40, 20:110, 20:168, 28:351
\propagate@doendpe 39:140, 811
\protectd@edf 1262
\protected 283
\protected@cmd 1226
\protected@edef .. 22:240, 29:546,
35:126, 35:136, 35:143, 35:151,
35:153, 35:174, 35:178, 35:185,
35:189, 35:196, 36:40, 36:45, 36:83,
36:84, 40:486, 40:503, 40:520, 44:60,
06:222, 45:472, 45:490, 45:508,
06:265, 50:539, 50:914, 50:1181, 1219
\protected@file@percent
..... 37:197, 37:204, 37:219,
37:227, 37:228, 44:165, 44:172, 766
\protected@wlog
..... 50:386, 50:388, 50:402, 50:416
\protected@write
... 20:202, 20:207, 35:105, 35:165,
36:55, 44:181, 46:14, 46:31, 1267
\protected@xdef 44:11,
45:453, 45:521, 45:537, 06:265,
50:378, 50:397, 50:450, 50:811, 1222
\provide@command 06:179, 06:178
\ps@empty 49:10, 57:158
\ps@plain 49:13
\q@curr@file 50:1209, 50:1253,
50:1255, 50:1260, 50:1286, 50:1384,
50:1386, 50:1391, 50:1419, 1253
\quote@@name 20:462, 20:478
\quote@name 20:462, 20:475,
20:477, 20:588, 20:590, 20:599, 50:1384
\r@@t 38:66
\reenable@package@load
..... 52:477, 53:464, 1055
\reinstall@nfss@defs 25:685, 25:726,
25:730, 25:732, 25:736, 25:737, 25:740
\relax 324
\rem@pt 24:329
\remove@angles 26:347, 26:370
\remove@nil 28:36
\remove@star 26:347, 26:353
\remove@tlig 21:1001,
21:1003, 21:1005, 21:1029, 21:1034,
21:1081, 21:1082, 21:1083, 1253
\remove@to@nnil
..... 24:328, 26:347, 26:373, 26:486
\renew@command
..... 06:125, 06:124, 06:185, 06:193
\renew@environment ... 06:153, 06:152
\requested@test@context
..... 29:503, 29:520, 29:524, 651
\reserved@@b 624

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

\reserved@a . . . . .  
 . . . . . 01:105, 03:13, 03:19, 03:34,  
 01:109, 01:110, 03:181, 03:182,  
 13:33, 13:37, 14:234, 18:456, 18:459,  
 20:212, 20:213, 20:235, 20:244,  
 20:253, 20:306, 20:368, 20:415,  
 20:485, 20:487, 20:492, 20:494,  
 20:506, 20:509, 20:512, 20:515,  
 20:516, 20:517, 20:539, 20:541,  
 20:546, 20:548, 20:549, 20:550,  
 20:561, 20:563, 20:568, 20:570,  
 20:585, 20:591, 20:595, 20:602,  
 20:608, 20:612, 20:641, 20:644,  
 20:645, 20:647, 20:656, 20:659,  
 20:666, 20:669, 20:691, 20:692,  
 20:693, 20:697, 20:705, 20:722,  
 20:723, 20:727, 20:733, 20:763,  
 20:767, 20:770, 20:775, 21:81,  
 21:82, 21:86, 21:89, 21:97, 21:107,  
 21:110, 21:119, 21:138, 21:143,  
 21:995, 21:999, 21:1043, 21:1045,  
 21:1046, 21:1048, 21:1050, 21:1058,  
 21:1067, 24:30, 24:31, 24:32, 24:41,  
 24:44, 24:47, 24:63, 24:66, 24:69,  
 24:105, 24:108, 24:110, 24:147,  
 24:151, 24:414, 24:417, 24:544,  
 24:545, 24:560, 24:563, 24:568,  
 24:579, 24:580, 24:593, 24:597,  
 24:602, 24:629, 24:632, 24:633,  
 24:641, 25:435, 25:436, 25:439,  
 25:440, 25:455, 25:456, 25:468,  
 25:469, 25:648, 25:649, 25:652,  
 25:653, 25:674, 25:675, 26:196,  
 26:198, 26:200, 26:210, 26:212,  
 26:215, 26:344, 26:345, 26:358,  
 26:359, 01:179, 01:180, 27:53, 27:57,  
 01:183, 28:460, 28:463, 28:535,  
 28:538, 28:571, 28:580, 28:582,  
 28:626, 28:629, 28:639, 28:642,  
 28:740, 28:742, 28:804, 28:808,  
 28:852, 28:854, 28:915, 28:918,  
 28:1010, 28:1012, 28:1106, 28:1108,  
 28:1210, 28:1212, 28:1228, 28:1230,  
 28:1231, 28:1236, 29:40, 29:41,  
 29:73, 29:74, 29:174, 29:175, 01:201,  
 29:516, 29:517, 01:205, 32:47, 32:48,  
 32:53, 32:54, 32:65, 32:68, 32:88,  
 32:95, 01:227, 05:22, 05:23, 05:24,  
 05:25, 05:26, 05:47, 05:52, 36:40,  
 36:41, 36:45, 36:46, 36:83, 36:85,  
 01:234, 05:89, 05:95, 05:105, 37:195,  
 37:196, 01:237, 37:250, 37:251,  
 37:256, 37:271, 37:272, 37:276,  
 37:289, 37:290, 37:294, 37:299,  
 37:300, 37:304, 37:389, 37:390,  
 01:239, 01:240, 01:247, 38:418,  
 38:419, 38:420, 38:421, 38:423,  
 01:250, 01:252, 01:253, 40:78, 40:79,  
 40:82, 40:97, 40:98, 40:101, 40:184,  
 40:190, 01:260, 41:241, 01:263,  
 41:245, 41:250, 41:269, 41:360,  
 41:361, 01:265, 42:199, 42:201,  
 06:117, 42:205, 06:120, 06:133,  
 06:134, 06:135, 06:137, 42:477,  
 42:478, 42:505, 42:506, 42:534,  
 42:535, 06:184, 06:185, 06:186,  
 06:192, 06:193, 06:194, 06:195,  
 06:198, 45:29, 45:30, 45:32, 45:33,  
 45:63, 45:67, 45:72, 45:74, 45:76,  
 45:78, 45:83, 45:84, 45:132, 45:136,  
 45:142, 45:145, 45:148, 06:222,  
 45:151, 06:228, 06:238, 06:242,  
 06:300, 06:304, 06:338, 06:342,  
 50:131, 50:134, 50:136, 06:366,  
 06:370, 50:230, 50:238, 50:242,  
 50:248, 50:256, 50:260, 50:448,  
 50:450, 50:451, 50:452, 50:456,  
 50:471, 50:473, 50:474, 50:475,  
 50:479, 50:656, 50:660, 50:666,  
 50:670, 50:748, 50:749, 50:752,  
 50:813, 50:817, 50:829, 50:830,  
 50:832, 50:841, 50:845, 50:857,  
 50:858, 50:860, 50:868, 50:872,  
 50:884, 50:885, 50:886, 50:888,  
 50:900, 50:903, 50:904, 50:906,  
 50:990, 50:1039, 50:1056, 50:1097,  
 50:1220, 50:1221, 50:1223, 50:1355,  
 50:1356, 50:1358, 50:1400, 50:1401,  
 50:1403, 50:1407, 06:481, 06:489,  
 50:1622, 50:1627, 06:490, 50:1679,  
 50:1680, 50:1711, 50:1712, 50:1783,  
 50:1784, 50:1808, 50:1810, 06:522,  
 01:315, 06:530, 06:531, 06:532,  
 06:533, 01:316, 01:317, 52:161,  
 52:167, 52:168, 52:169, 06:573,  
 06:576, 06:591, 06:593, 06:596,  
 06:605, 06:627, 06:634, 06:659,  
 06:661, 54:37, 54:46, 54:48, 54:50,  
 54:881, 54:901, 06:780, 06:789,  
 54:2001, 54:2003, 54:2004, 54:2093,  
 54:2095, 54:2101, 54:2104, 57:226,  
 57:243, 57:244, 57:245, 57:252,  
 57:253, 57:254, 57:489, 57:520,  
 57:526, 57:527, 57:538, 57:539,  
 57:540, 57:547, 57:548, 57:549,  
 57:574, 57:575, 57:579, 57:582,  
 57:584, 57:586, 57:591, 57:594,  
 57:602, 57:603, 57:604, 57:671,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

57:674, 57:675, 57:692, 02:193, 1221  
 \reserved@b ..... 01:106, 01:107,  
 13:33, 13:34, 13:37, 18:457, 18:458,  
 18:465, 20:304, 20:306, 20:366,  
 20:368, 20:413, 20:415, 20:586,  
 20:588, 20:590, 20:603, 20:605,  
 20:607, 20:691, 20:692, 20:693,  
 20:766, 20:768, 20:769, 20:776,  
 21:90, 21:97, 21:112, 21:119, 21:998,  
 21:999, 21:1044, 21:1045, 21:1067,  
 21:1076, 21:1078, 24:31, 24:32,  
 24:38, 24:95, 24:97, 24:150, 24:151,  
 24:630, 24:641, 25:454, 25:455,  
 25:457, 27:47, 27:54, 27:71, 27:73,  
 28:461, 28:462, 28:463, 28:467,  
 28:469, 28:536, 28:537, 28:538,  
 28:542, 28:544, 28:579, 28:580,  
 28:581, 28:616, 28:618, 28:697,  
 28:699, 28:744, 28:745, 28:746,  
 28:753, 28:913, 28:917, 28:919,  
 29:181, 29:186, 29:190, 29:191,  
 29:198, 29:199, 32:52, 32:53, 32:66,  
 32:68, 32:95, 32:96, 05:48, 05:54,  
 36:84, 36:85, 41:246, 41:248, 41:250,  
 06:109, 06:111, 06:118, 06:135,  
 06:136, 45:43, 45:44, 45:112, 45:113,  
 06:239, 06:240, 06:242, 06:301,  
 06:302, 06:304, 06:339, 06:340,  
 06:342, 06:367, 06:368, 06:370,  
 50:231, 50:232, 50:233, 50:235,  
 50:250, 50:253, 50:448, 50:471,  
 50:821, 50:827, 50:830, 50:849,  
 50:855, 50:858, 50:876, 50:882,  
 50:886, 50:900, 50:907, 50:1269,  
 50:1270, 50:1273, 50:1274, 50:1309,  
 50:1310, 50:1312, 50:1339, 50:1402,  
 50:1403, 50:1406, 50:1407, 50:1442,  
 50:1443, 50:1445, 50:1471, 50:1531,  
 50:1532, 50:1534, 50:1561, 06:574,  
 06:576, 06:592, 06:596, 06:631,  
 06:634, 54:790, 54:793, 54:807,  
 54:810, 54:827, 54:830, 06:781,  
 06:791, 57:229, 57:231, 57:235,  
 57:492, 57:494, 57:498, 57:692, 524  
 \reserved@c .....  
 ... 01:107, 01:112, 20:756, 24:96,  
 24:97, 24:631, 24:634, 27:48, 27:55,  
 27:61, 27:68, 28:33, 28:37, 28:468,  
 28:469, 28:543, 28:544, 28:617,  
 28:618, 28:698, 28:699, 28:721,  
 28:730, 28:745, 28:759, 28:1000,  
 28:1017, 28:1026, 28:1054, 28:1065,  
 28:1123, 28:1136, 28:1138, 29:183,  
 29:186, 29:196, 29:197, 29:200,  
 29:201, 32:67, 32:69, 32:76, 50:901,  
 50:903, 50:904, 50:1254, 50:1259,  
 50:1260, 50:1278, 50:1286, 50:1292,  
 50:1314, 50:1322, 50:1385, 50:1390,  
 50:1391, 50:1411, 50:1419, 50:1425,  
 50:1447, 50:1454, 50:1502, 50:1503,  
 50:1504, 50:1514, 50:1536, 50:1543,  
 50:1571, 06:786, 06:789, 06:791,  
 06:794, 57:233, 57:238, 57:246,  
 57:496, 57:517, 57:518, 57:519,  
 57:521, 57:522, 57:523, 57:524,  
 57:525, 57:533, 57:541, 57:694, 1235  
 \reserved@d .....  
 ... 01:110, 01:113, 20:754, 20:756,  
 27:61, 27:68, 27:70, 27:74, 28:1008,  
 28:1017, 28:1026, 28:1062, 28:1065,  
 28:1131, 28:1136, 28:1140, 29:188,  
 29:189, 29:194, 29:195, 05:46,  
 05:51, 06:779, 06:788, 57:695, 1251  
 \reserved@e .....  
 18:58, 18:60, 18:70, 18:72, 18:101,  
 18:108, 18:116, 27:39, 27:45, 27:70,  
 27:73, 27:74, 28:34, 28:39, 57:696, 1228  
 \reserved@f .....  
 18:59, 18:60, 18:71, 18:72, 18:116,  
 21:1543, 21:1544, 21:1545, 21:1546,  
 21:1548, 21:1559, 24:190, 24:192,  
 24:198, 24:199, 26:382, 26:393,  
 26:397, 26:401, 26:407, 26:410,  
 26:449, 26:486, 26:489, 27:27, 27:38,  
 27:45, 27:71, 27:73, 57:697, 1221  
 \reset@font . 29:219, 29:653, 29:684,  
 29:699, 29:714, 35:17, 35:34, 35:51,  
 40:482, 40:500, 40:517, 45:175,  
 45:371, 45:466, 45:485, 45:503,  
 47:40, 49:14, 54:620, 54:679, 1226  
 \restglob@settings .... 26:268, 26:278  
 \restore@mathversion .....  
 ... 28:107, 28:110, 28:125, 28:133  
 \restore@protect ..... 06:265  
 \rlh@ ..... 30:471, 30:472  
 \rm@def@ult ..... 29:397  
 \rmdef@ult ..... 29:258, 29:281,  
 29:319, 29:339, 29:355, 29:366,  
 29:398, 29:439, 33:27, 33:84, 643  
 \robust@command@act .....  
 ... 09:90, 06:422, 06:423,  
06:425, 06:493, 06:550, 06:687, 96  
 \robust@command@act@chk@args ...  
 ... 06:447, 06:468, 96  
 \robust@command@act@do .....  
 ... 06:433, 06:465, 97  
 \robust@command@act@end .. 06:430,  
 06:431, 06:443, 06:446, 06:466, 97

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\robust@command@act@loop 06:427, 06:433, 06:463, 97
\robust@command@act@loop@aux ... 06:433, 06:464
\robust@command@chk@safe 09:137, 06:315, 06:426,
 06:447, 06:467, 06:607, 06:619, 312
\s@fct@ 26:426, 26:490
\s@fct@alias 26:552
\s@fct@fixed 26:565
\s@fct@gen 26:502
\s@fct@genb 26:507
\s@fct@sgen 26:502
\s@fct@sgenb 26:507
\s@fct@sub 26:514
\s@fct@subf 26:557
\saved@space@catcode . 57:355, 57:424
\scan@fontshape ... 27:7, 27:40, 27:43
\scan@fontshape ... 27:6, 27:26, 27:37
\scantokens 320
\scriptfont@name 26:333, 26:338
\section 305
\select@group 24:614, 24:633,
 28:48, 28:387, 28:448, 28:626,
 28:679, 28:688, 28:726, 28:758, 607
\series@change@debug 29:137, 29:144, 29:147,
 29:158, 29:161, 29:165, 29:177,
 29:185, 29:190, 29:196, 29:199, 29:201
\series@check@toks 25:490, 25:492, 25:499
\series@drop@one@m 25:496, 25:500, 25:518
\series@maybe@drop@one@m 24:31, 25:483,
 25:485, 25:517, 28:462, 28:537,
 29:168, 29:187, 29:193, 29:401, 29:402
\series@maybe@drop@one@m@x 25:486, 25:488
\seriesdefault@kernel 29:220, 29:773, 660
\set@@mathdelimiter . 28:1063, 28:1097
\set@color 40:109
\set@curr@file ... 20:226, 20:235,
 20:262, 20:270, 20:452, 20:470,
 50:932, 50:1252, 50:1383, 52:266, 1002
\set@curr@file@aux 52:272, 52:278, 52:279
\set@curr@file@nosearch . 52:266, 1261
\set@display@protect 14:7, 14:14, 14:34,
 14:61, 06:8, 06:16, 06:263, 50:389, 1219
\set@fontsize 24:317,
 24:319, 26:121, 26:167, 26:177, 1211
\set@mathaccent 28:813, 28:821, 28:856, 28:864, 28:882
\set@mathchar 28:937, 28:947
\set@mathdelimiter 28:1014, 28:1023, 28:1075
\set@mathradical 28:395, 28:1133
\set@mathsymbol 28:921, 28:929, 28:950
\set@simple@size@args 26:348, 26:361, 26:368, 26:389, 26:403
\set@size@funct@args 26:351, 26:353, 26:411
\set@size@funct@args@ 26:411
\set@target@series 24:288,
 24:300, 25:437, 25:441, 25:444,
 25:447, 25:470, 25:472, 25:481, 25:516
\set@typeset@protect 41:197, 41:235,
 06:263, 06:282, 53:84, 53:128,
 54:607, 54:609, 54:665, 54:667, 1219
\SetMathAlphabet@ 28:633, 28:702, 28:711
\SetSymbolFont@ 28:493, 28:547, 28:565
\SetSymbolFont@m@dropped 28:535, 28:540, 28:560, 28:561
\sf@def@ult 29:397
\sf@size 19:6, 21:302, 24:224,
 24:243, 24:653, 26:328, 26:332,
 33:672, 45:409, 45:416, 45:437, 45:444
\sfdef@ult 29:259, 29:282,
 29:320, 29:340, 29:356, 29:367,
 29:399, 29:440, 33:29, 33:86, 643
\sh@ft 02:539, 1222
\show@kernel@robust@command
 . 06:613, 06:676, 06:704
\show@release@info 03:38, 03:41, 03:46, 03:51, 37:45,
 37:46, 37:48, 37:110, 37:111, 37:113, 38
>ShowCommand 312
\sixt@on 04:30, 24:14, 28:84, 28:248,
 28:799, 28:801, 28:847, 28:849,
 28:908, 28:910, 28:956, 28:958,
 28:996, 28:998, 28:1004, 28:1006,
 28:1050, 28:1052, 28:1058, 28:1060,
 28:1119, 28:1121, 28:1127, 28:1129,
 42:278, 42:293, 42:295, 45:62, 45:80,
 45:131, 45:153, 02:16, 54:1009,
 54:1055, 54:1194, 54:1362, 54:1596,
 54:1660, 54:1717, 54:1787, 54:2047,
 54:2056, 54:2112, 54:2128, 54:2161,
 02:64, 02:66, 02:96, 02:97, 02:98, 01:55
\sixt@on_ 382
\size@update 26:146, 26:172, 26:185, 26:204, 26:206
\sizefn@info 26:352, 26:354, 26:362, 26:390, 26:404

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\skip@ ..... 32:105, 32:108,  
02:41, 02:501, 02:503, 02:504, 02:506  
\sp@ce@skip ..... 18:84, 18:517, 18:518  
\sp@n ..... 41:379  
\split@name ..... 24:382,  
24:394, 24:508, 24:523, 26:519, 26:533  
\ssf@size .. 21:328, 21:360, 21:1254,  
24:225, 24:244, 24:654, 26:328, 26:334  
\string@makeletter 50:906, 50:941,  
50:942, 50:943, 52:169, 06:806, 1254  
\strip@meaning ..... 1204  
\strip@prefix .....  
... 01:95, 24:611, 01:212, 06:240,  
06:302, 06:340, 06:368, 01:307, 06:803  
\strip@pt ... 24:216, 24:222, 24:223,  
24:224, 24:225, 24:238, 24:242,  
24:329, 24:653, 24:654, 26:180, 02:543  
\sub@sfcnt 26:514, 26:515, 26:516, 26:543  
\subf@sfcnt .... 26:557, 26:558, 26:559  
\subst@correction ..... 24:85, 24:91  
\subst@fontshape ..... 27:8, 27:80  
\subst@size ..... 26:465  
\sw@slant ..... 32:91, 32:101  
\t@st@ic ..... 32:90, 32:94, 1199  
\target@meta@family@value .....  
..... 29:150, 29:175, 29:182, 29:184  
\target@series@value .....  
..... 29:149, 29:157, 29:160,  
29:162, 29:166, 29:167, 29:168,  
29:191, 29:197, 29:198, 29:200, 647  
\tc@check@accent ..... 33:109,  
33:161, 33:163, 33:165, 33:167,  
33:169, 33:171, 33:173, 33:175,  
33:177, 33:179, 33:181, 33:183,  
33:185, 33:187, 33:189, 33:191,  
33:935, 33:1011, 33:1013, 33:1015  
\tc@check@symbol .. 33:109, 33:211,  
33:213, 33:215, 33:217, 33:219,  
33:221, 33:223, 33:225, 33:227,  
33:229, 33:231, 33:233, 33:235,  
33:237, 33:239, 33:241, 33:243,  
33:245, 33:247, 33:249, 33:251,  
33:253, 33:255, 33:257, 33:259,  
33:261, 33:263, 33:265, 33:267,  
33:269, 33:271, 33:273, 33:275,  
33:277, 33:279, 33:281, 33:283,  
33:285, 33:287, 33:289, 33:291,  
33:293, 33:295, 33:297, 33:299,  
33:301, 33:303, 33:305, 33:307,  
33:310, 33:312, 33:314, 33:316,  
33:318, 33:320, 33:322, 33:324,  
33:326, 33:328, 33:330, 33:332,  
33:334, 33:336, 33:338, 33:340,  
33:342, 33:344, 33:346, 33:348,  
33:350, 33:354, 33:356, 33:358,  
33:360, 33:362, 33:364, 33:366,  
33:935, 33:1005, 33:1007, 33:1009,  
33:1017, 33:1019, 33:1021, 33:1023,  
33:1025, 33:1027, 33:1029, 33:1031,  
33:1033, 33:1035, 33:1037, 33:1039,  
33:1041, 33:1043, 33:1045, 33:1047,  
33:1049, 33:1051, 33:1053, 33:1055,  
33:1057, 33:1059, 33:1061, 33:1063,  
33:1065, 33:1067, 33:1069, 33:1071,  
33:1073, 33:1075, 33:1077, 33:1079,  
33:1081, 33:1083, 33:1085, 33:1087,  
33:1089, 33:1091, 33:1093, 33:1095,  
33:1097, 33:1099, 33:1101, 33:1103,  
33:1105, 33:1107, 33:1109, 33:1111,  
33:1113, 33:1115, 33:1117, 33:1119  
\tc@error ..... 33:110, 33:915, 33:936  
\tc@errorwarn ..... 33:21,  
33:76, 33:78, 33:823, 33:825, 33:827,  
33:828, 33:874, 33:875, 33:876, 33:909  
\tc@fake@euro .....  
..... 33:97, 33:352, 33:923, 33:1004  
\tc@forcedfalse ..... 33:868  
\tc@forcedtrue ..... 33:873  
\tc@oldstylesubst ..... 33:16, 33:20  
\tc@subst . 33:77, 33:109, 33:908, 33:935  
\tc@swap@accent ..... 33:112, 33:113  
\test@font@series@context .....  
..... 29:505, 29:515, 29:535, 651  
\test@next ..... 1227  
\text@command ..... 32:8, 32:46, 1216  
\textfont@name ..... 26:331, 26:337  
\tf@size ..... 24:223,  
24:243, 24:652, 26:328, 26:330, 1197  
\thr@@ ..... 02:563,  
02:573, 02:607, 26:58, 26:254,  
26:260, 26:273, 26:280, 26:287,  
26:292, 38:376, 38:539, 39:256,  
39:267, 42:287, 42:288, 42:290,  
42:291, 42:329, 42:355, 42:388,  
42:411, 02:16, 57:84, 57:92, 1224  
\toks@ 03:94, 03:106, 03:108, 03:115,  
03:119, 03:122, 03:127, 18:455,  
18:456, 18:461, 24:148, 24:152,  
24:154, 24:157, 24:221, 24:226, 28:6,  
28:7, 28:434, 28:438, 28:444, 28:447,  
28:452, 28:518, 28:519, 28:521,  
28:522, 28:572, 28:574, 28:578,  
28:595, 28:598, 28:657, 28:669,  
28:670, 28:671, 28:717, 28:719,  
28:725, 28:733, 28:737, 28:749,  
28:752, 28:755, 28:763, 28:765,  
28:1186, 28:1188, 28:1190, 28:1193,  
28:1195, 28:1198, 28:1201, 28:1233,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx

28:1234, 50:501, 50:502, 50:504,  
 50:505, 54:2247, 54:2248, 54:2249,  
 54:2250, 06:920, 06:921, 02:41, 1224  
 \topmark ..... 948  
 \topmark(s) ..... 948  
 \try@load@font@shape ..... 1236  
 \try@load@fontshape .....  
     ... 24:397, 24:405, 24:451, 24:556,  
     25:479, 26:520, 28:359, 28:376, 563  
 \try@simple@size ..... 26:356, 26:481  
 \try@simples ... 26:439, 26:445, 26:449  
 \try@size@orange 26:101, 26:356, 26:432  
 \try@size@substitution 26:103, 26:436  
 \tryif@simple ..... 26:447, 26:448  
 \tryis@simple ..... 26:448  
 \tt@def@ult ..... 29:397  
 \ttdef@ult ..... 29:260, 29:283,  
     29:321, 29:341, 29:357, 29:368,  
     29:400, 29:441, 33:31, 33:88, 643  
 \tw@ ..... 02:16, 1208  
 \two@digits ..... 01:169,  
     01:170, 26:512, 06:2, 50:1205,  
     50:1298, 50:1431, 50:1520, 01:70  
 \type@restoreinfo ..... 26:202, 26:207  
 \undeclare@... ..... 1084  
 \undeclare@file@substitution ...  
     ... 52:247, 1040  
 \unexpandable@noexpand ..... 1230  
 \unexpanded ..... 944  
 \unqu@tefilef@und ..... 52:143  
 \unquote@name .....  
     ... 20:456, 20:462, 20:479, 52:349  
 \unrestored@protected@xdef ...  
     ... 45:458, 45:526, 45:542, 45:553,  
     06:265, 48:223, 49:26, 49:54, 49:78  
 \unskip ..... 953  
 \update@series@target@value ...  
     ... 29:118, 29:152, 29:173, 29:244  
 \update@uclc@with@cyrillic ...  
     ... 21:1505, 21:1533, 21:1567, 21:1575  
 \upper@bound .....  
     ... 26:383, 26:384, 26:385, 26:398  
 \use@mathgroup ... 24:620, 24:638,  
     24:640, 26:299, 28:63, 28:92, 28:639,  
     28:741, 28:744, 28:1211, 28:1235, 1194  
 \UTF@two@octets@noexpand ..... 1261  
 \UTFviii@four@octets .....  
     ... 57:410, 57:415, 57:421  
 \UTFviii@four@octets@ 57:410, 57:421  
 \UTFviii@four@octets@combine 57:445  
 \UTFviii@four@octets@noexpand 57:451  
 \UTFviii@four@octets@string . 57:448  
 \UTFviii@invalid ..... 57:349, 57:442

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspage.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\XXX@argdef 1206
\z@ 02:311, 382
\z@skip 02:311, 414
\zap@space 20:262, 20:282,
 29:546, 47:49, 50:231, 50:453,
 50:476, 50:488, 50:655, 50:771,
 50:811, 50:829, 50:840, 50:857,
 50:867, 50:884, 50:916, 50:1218, 50:1353
\zref@labelbyprop 751
tex commands:
 \tex_afterassignment:D 53:49, 53:138, 07:2541
 \tex_aftergroup:D . 53:57, 53:144, 1068
 \tex_currentgrouplevel:D 10:33, 53:48, 53:56, 53:137, 53:143
 \tex_deadcycles:D 53:91
 \tex_endlinechar:D 09:447, 07:1857, 07:1955,
 07:1956, 07:1962, 07:1987, 1266
 \tex_escapechar:D 09:202, 09:277, 48:105,
 48:106, 48:108, 52:212, 52:226,
 07:1097, 07:1108, 07:1316, 07:1326,
 07:1856, 07:2415, 07:2441, 08:1164, 256
 \tex_everypar:D 16:20, 16:24, 16:29, 16:53, 16:57,
 16:61, 16:76, 16:76, 16:116, 16:118,
 16:128, 16:129, 16:180, 16:181, 1260
 \tex_gdef:D 09:192, 09:267, 317
 \tex_hskip:D 16:102
 \tex_indent:D 16:123
 \tex_interactionmode:D 48:103, 48:104, 48:109
 \tex_lastnodetype:D 16:101, 48:79, 415
 \tex_lastxpos:D 36:243
 \tex_lastypos:D 36:244
 \tex_lowercase:D 07:2238, 07:2421
 \tex_marks:D 48:232
 \tex_newlinechar:D 09:448
 \tex_noindent:D 16:27, 16:59, 16:133, 410
 \tex_par:D 16:18, 16:51, 16:104, 16:111,
 16:135, 16:176, 16:177, 16:178, 415
 \tex_parskip:D ... 16:25, 16:26, 16:58
 \tex_savepos:D 747
 \tex_setbox:D ... 48:110, 53:50, 53:139
 \tex_shipout:D .. 53:126, 53:380, 53:510
 \tex_splitbotmarks:D 48:129, 48:160, 956
 \tex_splitfirstmarks:D 48:141, 48:178, 48:185, 956
 \tex_splitmaxdepth:D 48:67
 \tex_unskip:D 16:97, 48:77, 414
 \tex_vbadness:D 48:68
 \tex_vfuzz:D 48:69
 \tex_vspliD 48:110, 955
 \tex_vss:D 53:334
 \tex_xdef:D 09:198, 09:273, 317
\text 1196
text commands:
 \text_case_exclude_arg_t1 . 57:621
 \text_case_switch:nnnn 57:624
 \text_declare_case_equivalent:Nn
 57:626
 \text_declare_lowercase_mapping:nn
 57:630
 \text_declare_lowercase_mapping:nnn
 57:631
 \text_declare_titlecase_mapping:nn
 57:637
 \text_declare_titlecase_mapping:nnn
 57:638
 \text_declare_uppercase_mapping:nn
 57:644
 \text_declare_uppercase_mapping:nnn
 57:645
 \text_lowercase:n 1262
\textacutedbl 21:915, 21:1153,
 33:234, 33:235, 33:721, 33:968, 33:1349
\textascendercompwordmark 21:854, 33:154, 33:685, 33:951, 33:1380
\textasciiaacute 21:965, 21:1114,
 33:236, 33:237, 33:722, 33:992, 33:1368
\textasciibreve 21:913, 21:1152,
 33:238, 33:239, 33:723, 33:965, 33:1346
\textasciicaron 21:914, 21:1151,
 33:240, 33:241, 33:724, 33:966, 33:1347
\textasciicircum 21:286, 21:558, 21:1088, 1246
\textasciidieresis .. 21:953, 21:1101,
 33:242, 33:243, 33:725, 33:982, 33:1365
\textasciigrave 21:904, 21:1082,
 33:244, 33:245, 33:726, 33:963, 33:1337
\textascimacron ... 21:960, 21:1109,
 33:246, 33:247, 33:727, 33:987, 33:1348
\textasciitilde 21:287, 21:559, 21:1093, 1230
\textasteriskcentered 21:267, 21:717, 21:864, 21:865,
 21:1215, 22:212, 22:218, 33:120,
 33:577, 33:642, 33:958, 33:1383, 1246
\textbackslash 21:268, 21:560, 21:718, 21:1087, 1223
\textbaht 21:939, 21:1156, 33:252, 33:253,
 33:731, 33:1098, 33:1099, 33:1359
\textbar 21:269, 21:561, 21:719, 21:1091, 1217

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\textbardbl 21:270,
 21:720, 21:919, 21:1170, 22:217,
 33:127, 33:365, 33:366, 33:578,
 33:650, 33:797, 33:971, 33:1264, 1239
\textbf 32:19, 36:238, 638
\textbigcircle 21:729,
 21:892, 21:1232, 33:254, 33:255,
 33:732, 33:1050, 33:1051, 33:1334, 1240
\textblank 21:729,
 .. 21:861, 21:1229, 33:345, 33:346,
 33:785, 33:1020, 33:1021, 33:1271
\textborn 21:905, 33:256, 33:257, 33:387,
 33:733, 33:1056, 33:1057, 33:1338
\textbraceleft 21:271,
 21:308, 21:562, 21:721, 21:1090, 1218
\textbraceright 21:272,
 21:309, 21:563, 21:722, 21:1092, 1217
\textbrokenbar 21:951,
 21:1099, 33:128, 33:651, 33:980, 33:1392
\textbullet 21:273, 21:723, 21:921, 21:1179,
 33:121, 33:579, 33:643, 33:973, 33:1387
\textcapitalcompwordmark 21:853, 33:153, 33:684, 33:950, 33:1379
\textcelsius 21:922, 21:1198, 33:129, 33:353,
 33:354, 33:652, 33:791, 33:974, 33:1265
\textcent 21:947,
 21:1095, 33:130, 33:653, 33:977, 33:1389
\textcentoldstyle 21:924,
 33:258, 33:259, 33:392, 33:395,
 33:734, 33:1072, 33:1073, 33:1352, 706
\textcircled ... 21:279, 21:283, 21:300,
 21:301, 21:730, 21:893, 33:155,
 33:156, 33:655, 33:671, 33:687,
 33:855, 33:1120, 33:1122, 33:1373, 1218
\textcircledP 21:958, 21:1200, 33:260, 33:261,
 33:735, 33:1114, 33:1115, 33:1367
\textcolonmonetary 21:926, 21:1191, 33:313, 33:314,
 33:765, 33:1074, 33:1075, 33:1295
\textcommaabove 21:353, 21:355, 21:369, 21:370,
 21:458, 21:459, 21:700, 21:701, 1243
\textcommabelow 21:324, 21:326, 21:332, 21:333,
 21:703, 21:704, 21:705, 21:706,
 21:707, 21:708, 21:709, 21:710,
 21:711, 21:712, 21:1252, 21:1455,
 21:1456, 21:1457, 21:1458, 1243
\textcompsubstdefault 33:34,
 33:39, 33:91, 33:618, 33:913, 1250
\textcompwordmark 21:290, 21:291, 21:564, 21:1158, 1217
\textcopyleft 21:956, 33:262, 33:263, 33:368,
 33:736, 33:1112, 33:1113, 33:1366
\textcopyright 21:283, 21:317, 21:954, 21:1102,
 33:131, 33:654, 33:983, 33:1394, 1235
\textcurrency 21:949, 21:1097,
 33:335, 33:336, 33:778, 33:850,
 33:854, 33:1008, 33:1009, 33:1284
\textdagger 21:275, 21:312, 21:725, 21:917,
 21:1177, 22:213, 22:219, 33:123,
 33:581, 33:645, 33:969, 33:1384, 1218
\textdaggerdbl 21:274, 21:313, 21:724, 21:918,
 21:1178, 22:214, 22:220, 33:122,
 33:580, 33:644, 33:970, 33:1385, 1218
\textdblyhyphen 21:876, 33:266, 33:267, 33:369,
 33:738, 33:1022, 33:1023, 33:1322
\textdblyhyphenchar 21:912, 33:264, 33:265, 33:370,
 33:737, 33:1068, 33:1069, 33:1345
\textdegree 21:961,
 21:1110, 33:132, 33:656, 33:988, 33:1398
\textdied 21:907, 33:268, 33:269, 33:388,
 33:739, 33:1060, 33:1061, 33:1340
\textdiscount 21:941, 21:1190, 33:270, 33:271,
 33:740, 33:1102, 33:1103, 33:1360
\textdiv 21:978, 21:1135,
 33:133, 33:657, 33:1002, 33:1407
\textdivorced 21:906, 21:1235, 33:272, 33:273,
 33:741, 33:1058, 33:1059, 33:1339
\textdollar 21:255, 21:307, 21:438,
 21:565, 21:798, 21:862, 21:1084,
 33:114, 33:115, 33:623, 33:625,
 33:956, 33:1128, 33:1130, 33:1381, 1217
\textdollaroldstyle 21:923,
 33:274, 33:275, 33:393, 33:394,
 33:742, 33:1070, 33:1071, 33:1351, 706
\textdong 21:935, 21:1195, 33:315, 33:316,
 33:766, 33:1092, 33:1093, 33:1298
\textdownarrow 21:903, 21:1210, 33:317, 33:318,
 33:767, 33:1054, 33:1055, 33:1294
\texteightoldstyle 21:886, 33:214, 33:215, 33:384,
 33:710, 33:1040, 33:1041, 33:1331
\textellipsis 21:296, 21:321, 21:1180

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\textemdash . . . . . 21:256, 21:407, 21:411,  
   21:566, 21:570, 21:785, 21:1162, 1217  
\textendash . . . . . 21:257, 21:408, 21:410,  
   21:567, 21:569, 21:786, 21:1161, 1217  
\textestimated . . . . . 21:942,  
   21:1206, 33:329, 33:330, 33:774,  
   33:853, 33:1006, 33:1007, 33:1288  
\texteuro . . . . . 21:976,  
   21:1196, 33:351, 33:352, 33:789,  
   33:851, 33:1003, 33:1004, 33:1274  
\textexcldown . . . . . 21:258, 21:412,  
   21:414, 21:571, 21:787, 21:1094, 1217  
\textfiguredash . . . . .  
   21:410, 21:569, 21:1160, 21:1166, 1256  
\textfiveoldstyle . . . . .  
   ... 21:883, 33:216, 33:217, 33:381,  
   33:711, 33:1034, 33:1035, 33:1328  
\textfloatsep . . . . . 54:732,  
   54:745, 54:2142, 54:2192, 54:2341  
\textflorin . . . . . 21:925, 21:1149,  
   33:333, 33:334, 33:777, 33:975, 33:1283  
\textfont . . . . . 26:337, 38:251  
\textfouroldstyle . . . . .  
   ... 21:882, 33:218, 33:219, 33:380,  
   33:712, 33:1032, 33:1033, 33:1327  
\textfraction . 54:1954, 54:1957, 54:1982,  
   54:1985, 54:2134, 54:2335, 1236  
\textfractionsolidus . . . . .  
   ... 21:877, 21:1187, 33:337, 33:338,  
   33:780, 33:959, 33:1277, 33:1509, 709  
\textgravedbl . . . . . 21:916, 21:1154,  
   33:248, 33:249, 33:728, 33:967, 33:1350  
\textgreater . . . . .  
   21:281, 21:572, 21:740, 21:1086, 1230  
\textguarani . . . . .  
   ... 21:929, 33:276, 33:277, 33:391,  
   33:743, 33:1080, 33:1081, 33:1354  
\textheight . . . . .  
   20:22, 20:23, 20:92, 20:93, 20:149,  
   20:150, 45:257, 45:258, 45:261,  
   45:287, 45:301, 53:380, 54:78,  
   54:227, 54:228, 54:276, 54:401,  
   54:449, 54:478, 54:649, 54:708,  
   54:767, 54:819, 57:156, 57:157, 1208  
\texthorizontalbar . . . . .  
   21:411, 21:570, 21:1163, 21:1168, 1256  
\texthyphen . . . . .  
   . 21:260, 21:417, 21:574, 21:789, 1217  
\texthyphenchar . . . . .  
   . 21:259, 21:416, 21:573, 21:788, 1223  
\textindent . . . . . 1224  
\textinterrobang . . . . .  
   . 21:933, 21:1186, 33:349, 33:350,  
   33:787, 33:1088, 33:1089, 33:1272  
\textinterrobangdown . . . . .  
   .. 21:934, 21:1236, 33:347, 33:348,  
   33:786, 33:1090, 33:1091, 33:1273  
\textit . . . . . 32:21  
\textlangle . . . . .  
   .. 21:888, 21:1227, 33:309, 33:310,  
   33:762, 33:1044, 33:1045, 33:1301  
\textlbrace . . . . . 1217  
\textlbrackdbl . . . . . 21:900, 33:210,  
   33:211, 33:389, 33:707, 33:961, 33:1335  
\textleaf 21:908, 33:278, 33:279, 33:373,  
   33:744, 33:1062, 33:1063, 33:1341  
\textleftarrow . . . . .  
   .. 21:859, 21:1207, 33:319, 33:320,  
   33:768, 33:1016, 33:1017, 33:1291  
\textlegacyasteriskcentered . . . . .  
   ..... 33:589, 33:800  
\textlegacybardbl . . . . . 33:589, 33:801  
\textlegacybullet . . . . . 33:589, 33:802  
\textlegacydagger . . . . . 33:589, 33:804  
\textlegacydaggerdbl . . . . . 33:589, 33:803  
\textlegacyparagraph . . . . . 33:589, 33:805  
\textlegacyperiodcentered 33:589, 33:806  
\textlegacysection . . . . . 33:589, 33:807  
\textless . . . . .  
   21:280, 21:575, 21:739, 21:1085, 1230  
\textlira 21:931, 21:1192, 33:321, 33:322,  
   33:769, 33:1084, 33:1085, 33:1297  
\textlnot . . . . . 21:957,  
   21:1107, 33:134, 33:658, 33:985, 33:1396  
\textlquill . . . . .  
   .. 21:945, 21:1188, 33:280, 33:281,  
   33:745, 33:1108, 33:1109, 33:1363  
\textmacron . . . . . 1237  
\textmarried . . . . .  
   .. 21:909, 21:1234, 33:282, 33:283,  
   33:746, 33:1064, 33:1065, 33:1342  
\textmd . . . . . 32:19  
\textmho 21:891, 21:1205, 33:284, 33:285,  
   33:747, 33:1048, 33:1049, 33:1333  
\textminus . . . . . 21:889, 21:1211, 33:343,  
   33:344, 33:783, 33:960, 33:1278, 711  
\textmu . . . . . 21:966, 21:1115,  
   33:341, 33:342, 33:782, 33:993, 33:1280  
\textmusicalnote . . . . .  
   .. 21:910, 21:1233, 33:286, 33:287,  
   33:748, 33:1066, 33:1067, 33:1343  
\textnaira . . . . .  
   .. 21:928, 21:1193, 33:288, 33:289,  
   33:749, 33:1078, 33:1079, 33:1353  
\textnineoldstyle . . . . .  
   ... 21:887, 33:220, 33:221, 33:385,  
   33:713, 33:1042, 33:1043, 33:1332

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphyphen.dtx, 57=ltfinal.dtx

```

\textrnonbreakinghyphen 21:409, 21:568, 21:1159, 21:1164, 1256
\textrnormal 32:15
\textrnumero
 .. 21:940, 21:1199, 33:331, 33:332,
 33:775, 33:1100, 33:1101, 33:1287
\textronekcentered
 .. 21:487, 21:698, 21:699, 1240
\textrtohm 21:899, 21:1204, 33:339,
 33:340, 33:781, 33:852, 33:1005, 33:1279
\textronehalf 21:974,
 21:1124, 33:135, 33:659, 33:999, 33:1404
\textroneoldstyle
 .. 21:879, 33:222, 33:223, 33:377,
 33:714, 33:1026, 33:1027, 33:1324
\textronequarter 21:973,
 21:1123, 33:136, 33:660, 33:998, 33:1403
\textronesuperior
 .. 21:970, 21:1118, 33:137, 33:355,
 33:356, 33:661, 33:792, 33:996, 33:1268
\textropenbullet
 .. 21:943, 21:1231, 33:290, 33:291,
 33:750, 33:1104, 33:1105, 33:1361
\textrordfeminine 21:305, 21:955,
 21:1103, 33:138, 33:662, 33:984, 33:1395
\textrordmasculine 21:306, 21:971,
 21:1119, 33:139, 33:664, 33:997, 33:1402
\textrOrMath
 .. 22:209, 22:212, 22:213, 22:214,
 22:215, 22:216, 22:217, 22:218,
 22:219, 22:220, 22:225, 22:232, 1242
\textrparagraph 21:276, 21:310, 21:726,
 21:967, 21:1116, 22:216, 33:124,
 33:582, 33:646, 33:994, 33:1400, 1218
\textrperiodcentered
 .. 21:277, 21:727, 21:968, 21:1117,
 33:125, 33:583, 33:647, 33:995, 33:1401
\textrpermill 1236
\textrperthousand 1236
\textrpertenthousand
 .. 21:492, 21:937, 21:1182,
 33:306, 33:307, 33:308, 33:759,
 33:1094, 33:1095, 33:1132, 33:1357, 704
\textrperthousand 21:490,
 21:920, 21:1181, 33:118, 33:119,
 33:639, 33:972, 33:1131, 33:1386, 1236
\textrpeso 21:930, 21:1197, 33:292, 33:293,
 33:751, 33:1082, 33:1083, 33:1355
\textrpilcrow
 .. 21:938, 33:294, 33:295, 33:386,
 33:752, 33:1096, 33:1097, 33:1358
\textrpm 21:962,
 21:1111, 33:140, 33:666, 33:989, 33:1399
\textrquestiondown 21:261, 21:413,
 21:415, 21:576, 21:790, 21:1126, 1240
\textrquotedbl 21:579, 21:1083, 1217
\textrquotedblleft 21:262,
 21:418, 21:577, 21:791, 21:1174, 1217
\textrquotedblrigh 21:263,
 21:419, 21:578, 21:792, 21:1175, 1217
\textrquotaleft 21:264,
 21:420, 21:580, 21:793, 21:1171, 1217
\textrquoteright 21:265,
 21:421, 21:581, 21:794, 21:1172, 1217
\textrquotesingle 21:863,
 21:1081, 33:141, 33:667, 33:957, 33:1382
\textrquotestraightbase 21:855,
 33:142, 33:371, 33:668, 33:952, 33:1377
\textrquotestraightdblbase 21:856,
 33:143, 33:372, 33:669, 33:953, 33:1378
\textrtriangle
 .. 21:890, 21:1228, 33:311, 33:312,
 33:763, 33:1046, 33:1047, 33:1302
\textrbrace 1217
\textrbrackdbl 21:901, 33:212,
 33:213, 33:390, 33:708, 33:962, 33:1336
\textrrecipe
 .. 21:932, 21:1201, 33:296, 33:297,
 33:753, 33:1086, 33:1087, 33:1356
\textrreferencemark
 .. 21:969, 21:1185, 33:298, 33:299,
 33:754, 33:1116, 33:1117, 33:1369
\textrregistered
 .. 21:300, 21:301, 21:959, 21:1108,
 33:144, 33:670, 33:986, 33:1397, 1230
\textrrightarrow
 .. 21:860, 21:1209, 33:323, 33:324,
 33:770, 33:1018, 33:1019, 33:1292
\textrrm 32:15
\textrquill
 .. 21:946, 21:1189, 33:300, 33:301,
 33:755, 33:1110, 33:1111, 33:1364
\textrsc 32:21, 1235
\textrsection 21:278,
 21:311, 21:582, 21:728, 21:952,
 21:1100, 22:215, 33:126, 33:584,
 33:585, 33:648, 33:981, 33:1393, 1218
\textrservicemark
 .. 21:944, 21:1202, 33:302, 33:303,
 33:756, 33:1106, 33:1107, 33:1362
\textrsevenoldstyle
 .. 21:885, 33:224, 33:225, 33:383,
 33:715, 33:1038, 33:1039, 33:1330
\textrsf 32:15
\textrsixoldstyle
 .. 21:884, 33:226, 33:227, 33:382,
 33:716, 33:1036, 33:1037, 33:1329

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=ltphypen.dtx, 57=ltfinal.dtx

\textsl ..... 32:21  
 \textssc ..... 25:536, 32:25  
 \textsterling .. 21:266, 21:319, 21:445,  
     21:583, 21:805, 21:948, 21:1096,  
     33:116, 33:117, 33:624, 33:632,  
     33:978, 33:1127, 33:1129, 33:1390, 1217  
 \textstyle ..... 19:15, 30:476, 38:63  
 \textsubscript ..... 45:419  
 \textsuperscript 21:303, 21:305, 21:306,  
     33:663, 33:665, 33:679, 45:402, 1225  
 \textsurd 21:972, 21:1226, 33:304, 33:305,  
     33:757, 33:1118, 33:1119, 33:1370  
 \textsw ..... 25:531, 32:25  
 \TextSymbolUnavailable 21:3, 21:758, 1230  
 \textthreeoldstyle .....  
     ... 21:881, 33:228, 33:229, 33:379,  
     33:717, 33:1030, 33:1031, 33:1326  
 \textthreequarters .. 21:975, 21:1125,  
     33:146, 33:675, 33:1000, 33:1405  
 \textthreequartersemdash .....  
     ... 21:858, 33:145, 33:357, 33:358,  
     33:375, 33:674, 33:793, 33:955, 33:1263  
 \textthreesuperior .....  
     ... 21:964, 21:1113, 33:147, 33:359,  
     33:360, 33:676, 33:794, 33:991, 33:1267  
 \texttildelow ..... 21:911, 21:1155,  
     33:250, 33:251, 33:729, 33:964, 33:1344  
 \texttimes ..... 21:977, 21:1129,  
     33:148, 33:677, 33:1001, 33:1406  
 \texttrademark 21:303, 21:936, 21:1203,  
     33:149, 33:678, 33:976, 33:1388, 1230  
 \texttt ..... 32:15  
 \texttwelveudash .....  
     ... 21:857, 33:150, 33:361, 33:362,  
     33:374, 33:680, 33:795, 33:954, 33:1262  
 \texttwooldstyle .....  
     ... 21:880, 33:230, 33:231, 33:378,  
     33:718, 33:1028, 33:1029, 33:1325  
 \texttwosuperior .....  
     ... 21:963, 21:1112, 33:151, 33:363,  
     33:364, 33:681, 33:796, 33:990, 33:1266  
 \textulc ..... 25:526, 32:25  
 \textunderline ..... 1217  
 \textunderscore .....  
     21:288, 21:315, 21:584, 21:1089, 1217  
 \textup ..... 32:21, 566  
 \textuparrow .....  
     ... 21:902, 21:1208, 33:325, 33:326,  
     33:771, 33:1052, 33:1053, 33:1293  
 \textvisibleSPACE .....  
     ... 21:292, 21:585, 21:1230, 1217  
 \textwidth 20:24, 20:94, 20:151, 40:444,  
     45:266, 54:79, 54:146, 54:203,  
     54:220, 54:634, 54:644, 54:693,  
     54:703, 54:2261, 54:2293, 57:157, 1208  
 \textwon 21:927, 21:1194, 33:327, 33:328,  
     33:772, 33:1076, 33:1077, 33:1296  
 \textyen ..... 21:950,  
     21:1098, 33:152, 33:682, 33:979, 33:1391  
 \textzerooldstyle .....  
     ... 21:878, 33:232, 33:233, 33:376,  
     33:719, 33:1024, 33:1025, 33:1323  
 \TH ..... 21:535, 21:1131, 57:650, 1213  
 \th ..... 21:586, 21:1137, 57:650, 1213  
 \thanks ..... 898  
 \thanks ..... 44:10, 44:26, 1219  
 \the ..... 412  
 \the... ..... 1267  
 \the#1 ..... 514  
 thebibliography (env.) ..... 937  
 \theenum ..... 816  
 \theequation 38:352, 38:364, 38:457, 38:518  
 \thefootnote 45:396, 45:521, 45:526, 45:546  
 \thempfn ..... 40:454,  
     45:453, 45:458, 45:537, 45:542, 45:545  
 \thempfootnote ..... 40:454, 45:398  
 \thepage ..... 20:209, 34:6, 35:18,  
     35:35, 35:52, 35:106, 35:166, 36:211,  
     36:222, 36:227, 36:234, 44:164,  
     44:171, 44:177, 46:15, 46:32, 47:43,  
     49:14, 54:246, 54:277, 54:1831, 750  
 \Theta ..... 30:306  
 \theta ..... 30:282  
 \thetotalpages ..... 53:350, 53:442, 1080  
 \thicklines ..... 42:124  
 \thickmuskip 30:653, 38:228, 38:230, 38:243  
 \thickspace ..... 38:214  
 \thinlines ..... 42:124, 42:815, 42:832  
 \thinmuskip .....  
     30:651, 38:220, 38:222, 38:238, 38:244  
 \thinspace ..... 18:536, 18:542,  
     18:543, 38:189, 38:214, 38:251, 1251  
 \thispagestyle ..... 49:6  
 \tilde ..... 18:472, 18:483, 30:526  
 \time ..... 01:163, 01:167, 02:383  
 \times ..... 30:403  
 \title ..... 898  
 \title ..... 44:6, 44:7, 44:21, 44:23, 44:31  
 title ..... 36:237, 751  
 tl commands:  
     \c\_empty\_tl ..... 08:2194, 08:65  
     \c\_novalue\_tl 07:525, 07:909, 07:965,  
         07:1026, 07:1502, 07:1602, 07:1718,  
         07:1823, 07:1968, 07:1975, 07:2034,  
         07:2086, 07:2167, 07:2194, 07:2276, 235  
     \c\_space\_tl . 09:365, 11:543, 11:547,  
         11:590, 11:788, 11:791, 16:42,

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

16:73, 16:82, 52:553, 52:556, 07:24,
57:600, 57:605, 57:610, 57:615,
07:120, 07:136, 07:139, 07:153,
07:164, 07:165, 07:181, 07:189,
07:199, 07:202, 07:204, 07:206,
07:208, 07:216, 07:222, 07:290,
07:291, 07:1152, 07:1167, 07:1199,
07:1200, 07:1377, 07:1384, 07:1510
\tl_clear:N 09:196, 09:271,
11:284, 11:323, 11:706, 48:148,
48:149, 07:305, 07:350, 07:379,
07:446, 07:447, 07:663, 07:777,
07:788, 07:789, 07:790, 07:792,
07:950, 07:1018, 07:1212, 07:1445,
07:1446, 07:1518, 07:1539, 07:1629,
07:1630, 07:1646, 07:1858, 07:2226
\tl_const:Nn ... 09:11, 09:12, 11:9,
11:10, 11:11, 11:12, 11:13, 11:14,
11:15, 53:311, 53:316, 07:1468,
07:2546, 08:35, 08:36, 08:110,
08:113, 08:919, 08:920, 08:922,
08:923, 08:924, 08:925, 08:928,
08:929, 08:931, 08:952, 08:953, 08:1297
\tl_count:N ... 52:546, 07:397, 07:2250
\tl_count:n 07:602, 07:607, 07:864, 07:971,
07:1025, 07:1252, 07:1402, 07:1410
\tl_gclear:N 282
\tl_gclear_new:N . 36:25, 36:145, 08:270
\tl_gput_left:Nn 51:235
\tl_gput_right:Nn 16:38,
16:69, 16:78, 28:320, 48:173, 48:182,
07:1127, 07:1438, 07:1440, 08:455, 265
\tl_gremove_once:Nn 08:37, 08:37
\tl_gset:Nn 09:124, 16:15, 16:48, 17:34,
17:38, 28:284, 28:317, 36:26, 36:138,
36:146, 48:128, 48:139, 48:159,
48:208, 48:209, 48:210, 52:51,
55:105, 55:106, 55:126, 55:130,
55:176, 55:184, 55:230, 08:182,
08:194, 08:301, 08:319, 08:417,
08:429, 08:445, 08:484, 08:749,
08:803, 08:837, 08:841, 08:864, 08:868
\tl_gset_eq>NN 48:126, 48:127, 48:132,
48:134, 48:138, 48:197, 48:199,
48:201, 48:432, 48:434, 48:436,
48:438, 48:444, 48:448, 48:451,
55:207, 55:208, 08:63, 08:452, 08:463
\tl_head:N 07:2478
\tl_head:w 11:79, 11:92
\tl_if_blank:nTF 11:331, 11:335, 11:435, 11:438,
20:521, 55:53, 57:580, 07:598,
07:1612, 07:2048, 07:2089, 07:2300,
07:2989, 07:2990, 07:2991, 08:435
\tl_if_blank_p:n 07:116
\tl_if_empty:N 289
\tl_if_empty:NTF . 08:2177, 08:2179,
11:245, 11:258, 11:311, 48:130,
48:161, 48:235, 07:316, 07:317,
07:465, 07:713, 07:1452, 08:235,
08:255, 08:366, 08:368, 08:637,
08:1038, 08:1428, 08:1988, 08:1996
\tl_if_empty:nTF . 08:2503, 09:120,
09:305, 09:324, 11:1148, 11:1153,
49:30, 49:40, 52:40, 52:46, 52:58,
52:107, 52:114, 52:116, 52:118,
52:416, 07:2045, 07:2416, 07:2442,
07:2477, 07:2669, 07:2759, 08:338,
08:353, 08:356, 08:824, 08:856, 08:910
\tl_if_empty_p:N 08:2453, 08:2454, 53:68, 53:114, 53:378
\tl_if_empty_p:n 08:887
\tl_if_eq:NNTF 48:267, 48:274, 48:441, 07:339
\tl_if_eq:NnTF 36:151
\tl_if_eq:nnTF . 07:725, 07:984, 07:2421
\tl_if_exist:N 283
\tl_if_exist:NTF . 08:2222, 08:2242,
08:2328, 08:2476, 08:2517, 36:106,
36:111, 36:130, 36:172, 36:187,
07:1453, 08:135, 08:233, 08:253, 08:1275
\tl_if_exist_p:N 08:276
\tl_if_head_eq_charcode:nNTF ... 20:523, 20:528
\tl_if_head_is_group:ntf ... 09:332, 07:2298, 07:2342, 07:2374
\tl_if_head_is_N_type:nTF . 09:329,
07:2310, 07:2322, 07:2339, 07:2371
\tl_if_in:NnTF 07:1643
\tl_if_in:nnTF 11:280, 11:304, 11:326,
11:738, 07:614, 07:1609, 07:1635, 170
\tl_if_novalue:nTF 07:370, 07:389, 07:436, 07:954,
07:2118, 07:2981, 07:2982, 07:2983,
07:2984, 07:2985, 07:2986, 08:332
\tl_if_single:nTF 11:486, 07:2215, 07:2964
\tl_if_single_token:nTF 09:315, 07:673, 07:2413, 07:2966, 191
\tl_log:n 08:37, 08:39, 08:1821
\tl_map_function:Nn . 07:97, 07:390,
07:599, 07:600, 07:969, 07:1019, 07:3005
\tl_map_inline:Nn 07:723
\tl_map_inline:nn ... 07:1481, 07:1715

```

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\tl_new:N 09:6, 09:8, 09:9,
 09:10, 11:19, 11:20, 11:23, 11:24,
 11:25, 11:26, 11:27, 11:37, 16:14,
 36:137, 48:24, 48:25, 48:26, 48:27,
 48:28, 48:29, 48:30, 48:31, 48:32,
 48:33, 48:34, 48:35, 48:36, 48:37,
 48:38, 48:39, 48:40, 48:41, 48:63,
 48:64, 48:116, 48:192, 48:193, 52:8,
 52:9, 52:10, 52:11, 52:59, 53:53,
 53:205, 55:102, 55:103, 55:107,
 55:108, 55:109, 55:110, 07:11, 07:12,
 07:13, 07:14, 07:17, 07:21, 07:28,
 07:29, 07:30, 07:33, 07:38, 07:39,
 07:41, 07:42, 07:50, 07:51, 07:1624,
 07:1625, 07:1841, 07:2212, 07:3000,
 08:25, 08:26, 08:27, 08:29, 08:32,
 08:78, 08:96, 08:124, 08:138, 08:141,
 08:165, 08:166, 08:299, 08:317,
 08:1294, 08:1550, 08:1551, 08:1552
\tl_put_left:Nn 07:830, 07:839, 07:1537
\tl_put_right:Nn
 ... 09:168, 09:182, 09:243, 09:257,
 11:299, 11:306, 11:770, 11:778,
 11:805, 11:816, 11:833, 11:858,
 11:865, 57:621, 07:365, 07:391,
 07:535, 07:568, 07:591, 07:605,
 07:622, 07:627, 07:755, 07:855,
 07:868, 07:895, 07:903, 07:917,
 07:919, 07:926, 07:943, 07:959,
 07:965, 07:1034, 07:1059, 07:1227,
 07:1235, 07:1251, 07:1258, 07:1513,
 07:1516, 07:1633, 07:1634, 07:1641,
 07:1651, 07:1719, 07:1753, 07:1982,
 07:1998, 07:2009, 07:2228, 07:2273
\tl_replace_all:Nnn .. 11:279, 07:2241
\tl_rescan:nn
 ... 09:20, 09:20, 09:462, 09:538, 09:566
\tl_set:Nn
 ... 09:166, 09:179, 09:180, 09:183,
 09:185, 09:206, 09:241, 09:254,
 09:255, 09:258, 09:260, 09:281,
 09:373, 09:378, 09:459, 09:504,
 09:535, 09:563, 11:255, 11:278,
 11:322, 11:333, 11:337, 11:416,
 11:542, 11:546, 11:589, 11:620,
 11:727, 11:745, 11:876, 11:888,
 11:910, 48:42, 48:43, 48:44, 48:45,
 48:46, 48:47, 48:48, 48:49, 48:50,
 48:51, 48:52, 48:53, 48:54, 48:55,
 48:56, 48:57, 48:58, 48:59, 48:101,
 52:81, 52:82, 52:83, 52:84, 53:47,
 53:136, 53:210, 53:236, 53:261,
 53:288, 55:42, 55:112, 55:113,
 55:114, 55:115, 55:159, 55:160,
 07:22, 07:88, 07:119, 07:306, 07:307,
 07:308, 07:309, 07:337, 07:388,
 07:874, 07:1071, 07:1089, 07:1291,
 07:1509, 07:1544, 07:1606, 07:1712,
 07:1739, 07:1746, 07:1759, 07:1765,
 07:1771, 07:1777, 07:1783, 07:1789,
 07:1795, 07:1801, 07:1832, 07:1854,
 07:1877, 07:2208, 07:2209, 07:2235,
 07:2259, 07:2290, 07:2302, 07:2329,
 07:2395, 07:2397, 07:2516, 07:2517,
 08:37, 08:40, 08:557, 08:1148,
 08:1155, 08:1159, 08:1559, 08:1579,
 08:1580, 08:1585, 08:1586, 08:1601,
 08:1608, 08:1610, 08:1641, 08:1662,
 08:1663, 08:1669, 08:1671, 08:1690,
 08:1697, 08:1699, 08:1746, 08:1753
\tl_set_eq:NN 09:197, 09:272,
 48:153, 48:154, 55:43, 55:156,
 55:161, 07:327, 07:333, 07:341,
 07:384, 07:1538, 07:1645, 07:1718,
 08:1589, 08:1613, 08:1674, 08:1702
\tl_show:n 07:1333,
 07:1346, 07:1355, 07:1391, 07:1414,
 08:37, 08:38, 08:1826, 08:1929, 08:2030
\tl_tail:N 07:1911
\tl_to_str:N 07:1966, 07:1973, 07:2554
\tl_to_str:n 09:113,
 09:118, 09:457, 09:476, 09:505,
 09:516, 09:552, 11:302, 11:417,
 11:621, 11:876, 11:943, 11:944,
 11:945, 11:954, 11:955, 36:10, 36:19,
 36:51, 36:64, 36:92, 36:93, 36:100,
 36:101, 36:145, 36:146, 36:187,
 48:230, 48:245, 51:121, 51:150,
 51:152, 52:580, 07:70, 07:81, 07:248,
 07:252, 07:460, 07:468, 07:481,
 07:492, 07:496, 07:532, 07:563,
 07:644, 07:668, 07:676, 07:691,
 07:704, 07:716, 07:735, 07:772,
 07:773, 07:1225, 07:1299, 07:1305,
 07:1513, 07:1517, 07:1967, 07:1974,
 07:2162, 07:2189, 07:2269, 07:2270,
 07:2430, 07:2434, 08:384, 08:534,
 08:607, 08:1977, 08:1990, 08:2061, 1032
\tl_trim_spaces:n . 09:307, 11:301,
 11:322, 11:334, 11:417, 11:441,
 11:621, 11:876, 50:514, 51:111,
 51:142, 52:41, 52:42, 07:244, 07:417,
 07:535, 07:569, 07:1527, 07:1529,
 07:2161, 07:2188, 07:2290, 07:2844,
 07:2848, 07:2852, 07:2856, 08:413, 318
\tl_trim_spaces_apply:nN . 07:673,
 07:2293, 07:2317, 07:2407, 08:334
\tl_use:N ... 28:288, 28:292, 28:327,
```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrord.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

28:331, 36:94, 36:107, 07:1370,  
 07:1458, 08:1056, 08:1119, 08:1812  
 $\backslash l_{tmpa\_tl}$  ..... 346  
 tl internal commands:  
 $\backslash g_{\_mark\_first\_marks\_tl}$  .....  
 .. 48:148, 48:153, 48:182, 48:192, 958  
 $\backslash g_{\_mark\_last\_marks\_tl}$  .....  
 .. 48:149, 48:154, 48:173, 48:192, 958  
 $\backslash g_{\_mark\_new\_top\_tl}$  .....  
 . 48:61, 48:126, 48:127, 48:133, 48:135  
 $\backslash l_{\_mark\_saved\_parameters\_tl}$  ...  
 ..... 48:101, 48:114, 48:116  
 $\backslash g_{\_mark\_tmp\_tl}$  ..... 48:61,  
 48:128, 48:130, 48:138, 48:159,  
 48:161, 48:171, 48:174, 48:223,  
 48:230, 48:235, 48:237, 48:238, 960  
 $\backslash tmspace$  ..... 38:214, 791  
 $\backslash to$  ..... 30:447, 30:449  
 $\backslash today$  ..... 01:168,  
 01:172, 01:180, 01:183, 44:33, 1189  
 token commands:  
 $\backslash c_{space\_token}$  ..... 07:2506  
 $\backslash token\_if\_active:NTF$  07:1984, 07:2000  
 $\backslash token\_if\_cs:NTF$  ..... 07:2217, 180  
 $\backslash token\_if\_eq\_catcode:NNTF$  .....  
 ..... 09:317, 07:1663  
 $\backslash token\_if\_eq\_charcode:NNTF$  ....  
 .. 07:412, 07:1899, 07:1931, 07:1942  
 $\backslash token\_if\_eq\_meaning:NNTF$  .....  
 .. 09:187, 09:262, 09:463, 52:454,  
 52:466, 07:682, 07:685, 07:693,  
 07:1292, 07:2520, 07:2968, 07:2970  
 $\backslash token\_if\_long\_macro_p:N$  ... 07:1386  
 $\backslash token\_if\_macro:NTF$  .....  
 .. 09:84, 08:1204, 08:1230, 311  
 $\backslash token\_if\_math\_toggle_p:N$  .....  
 .. 07:2363, 07:2385  
 $\backslash token\_if\_protected\_long\_macro\_p:N$  ..... 07:1387, 07:1396  
 $\backslash token\_if\_protected\_macro:NTF$  ...  
 .. 07:1155, 07:1180, 07:1347  
 $\backslash token\_if\_protected\_macro_p:N$  07:1395  
 $\backslash token\_to\_meaning:N$  09:80, 09:460,  
 09:472, 09:536, 09:564, 16:41, 16:72  
 $\backslash token\_to\_str:N$  .....  
 .. 09:79, 09:80, 09:102, 09:522,  
 09:529, 09:557, 09:560, 11:279,  
 11:280, 11:297, 11:304, 11:307,  
 11:314, 11:1079, 36:57, 52:241,  
 07:70, 07:78, 07:81, 07:417, 07:808,  
 07:1066, 07:1220, 07:1369, 07:1393,  
 07:1416, 07:1676, 07:1822, 07:1989,  
 07:2001, 07:2161, 07:2188, 07:2417,  
 07:2430, 07:2434, 07:2443, 07:2811,  
 07:2812, 07:2825, 07:2826, 07:2930,  
 07:2931, 07:2944, 07:2945, 08:381, 160  
 $\backslash toks$  ..... 16:40, 16:71,  
 16:80, 04:36, 28:668, 28:669, 28:679,  
 28:688, 02:31, 57:698, 02:63, 02:95, 412  
 $\backslash toksdef$  ..... 04:230, 02:46, 02:63, 02:95  
 $\backslash tokszero$  ..... 04:230  
 $\backslash tolerance$  ..... 24:672,  
 24:718, 24:733, 49:87, 49:95, 02:317  
 $\backslash top$  ..... 30:327  
 $\backslash topfigrule$  ..... 54:731, 54:2363  
 $\backslash topfraction$  ..... 45:273, 54:2329  
 $\backslash topmargin$  ..... 54:71, 54:628, 54:687  
 $\backslash TopMark$  ..... 48:366, 48:486, 947  
 $\backslash topmark$  ..... 54:2247, 54:2256, 947  
 $\backslash topsep$  ..... 38:524, 39:1, 39:59, 803  
 $\backslash topskip$  ..... 20:59,  
 20:128, 20:183, 54:130, 02:418, 1227  
 $\backslash totalheight$  ..... 40:33, 40:34, 40:35  
 $\backslash totalpages$  ..... 1064  
 trace commands:  
 $\backslash trace\_stack\_levels$  ..... 57:96  
 $\backslash tracefloats$  ..... 54:1933  
 $\backslash tracefloatsoff$  ..... 54:1933  
 $\backslash tracefloatvals$  ..... 54:1933, 1258  
 $\backslash traceoff$  ..... 398  
 $\backslash traceon$  ..... 398  
 $\backslash tracingall$  ..... 02:557, 398  
 $\backslash tracingassigns$  .....  
 ..... 02:575, 02:609, 02:626, 02:667  
 $\backslash tracingcommands$  02:573, 02:591, 02:607,  
 02:616, 02:629, 02:653, 02:670, 02:347  
 $\backslash tracingfonts$  ..... 26:17, 26:54,  
 26:58, 26:86, 26:118, 26:153, 26:194,  
 26:224, 26:238, 26:254, 26:260,  
 26:273, 26:280, 26:287, 26:292,  
 26:301, 26:314, 26:322, 26:325, 1194  
 $\backslash tracinggroups$  .....  
 .. 02:565, 02:600, 02:638, 02:678  
 $\backslash tracingifs$  02:566, 02:601, 02:637, 02:677  
 $\backslash tracinglostchars$  .. 02:563, 02:586,  
 02:598, 02:617, 02:641, 02:661,  
 02:681, 24:682, 02:346, 02:558, 1258  
 $\backslash tracingmacros$  .. 02:572, 02:590, 02:606,  
 02:618, 02:640, 02:660, 02:680, 02:341  
 $\backslash tracingnesting$  .....  
 .. 02:568, 02:603, 02:635, 02:675  
 $\backslash tracingnone$  ..... 02:622, 1267  
 $\backslash tracingoff$  ..... 26:118, 26:322  
 $\backslash tracingon$  ..... 26:119, 26:323, 1195  
 $\backslash tracingonline$  .. 02:628, 02:652, 02:669,  
 02:699, 24:681, 54:1919, 02:340, 02:552  
 $\backslash tracingoutput$  ..... 02:632,  
 02:656, 02:673, 02:696, 02:345, 02:553

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterrordtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\tracingpages . . 02:562, 02:585, 02:597,
 02:617, 02:642, 02:662, 02:682, 02:344
\tracingparagraphs
 02:564, 02:587, 02:599,
 02:618, 02:639, 02:659, 02:679, 02:343
\tracingrestores 02:574, 02:592, 02:608,
 02:618, 02:627, 02:658, 02:668, 02:348
\tracingscantokens 02:567,
 02:582, 02:602, 02:636, 02:650, 02:676
\tracingstacklevels
 02:570, 02:634, 02:349, 02:558, 24
\tracingstats
 02:584, 02:596, 02:616, 02:643,
 02:663, 02:683, 57:2, 02:342, 02:561
\triangle 30:329
\triangleleft 30:365, 30:489
\triangleright 30:366, 30:489
\TrimSpaces 07:3001
trivlist (env.) 39:89
\trivlist 37:444, 37:494, 37:496,
 37:527, 37:549, 38:508, 39:89,
 41:78, 43:52, 43:54, 43:59, 43:61, 810
TS1 commands:
 \TS1:? 728
 \TS1:<family> 728
\tt 1199
\ttdefault 29:12, 29:216, 29:400,
 29:417, 29:441, 29:460, 29:488, 30:58
\ttfamily 29:10,
 29:11, 29:410, 29:458, 29:459,
 29:486, 29:487, 32:17, 37:571, 221
ttfamily 29:420
\ttsubstdefault 30:20, 30:32, 33:32, 33:89
\twocolumn 54:201, 1233
\twocolumn[...] 1228
\twocolumn[] 446
\typein 82, 82
\typein 33:1620, 33:1625, 06:31
\typeout 82
\typeout 01:100,
 03:21, 10:54, 10:57, 10:59, 10:61,
 10:63, 10:66, 10:69, 03:38, 14:74,
 20:201, 20:673, 20:674, 20:680,
 20:714, 20:760, 20:774, 20:795,
 01:156, 24:391, 25:478, 01:181,
 01:183, 01:195, 29:138, 29:563,
 29:744, 29:754, 29:764, 30:9, 30:132,
 01:210, 01:217, 33:1216, 33:1253,
 33:1260, 33:1413, 33:1416, 33:1419,
 33:1617, 33:1618, 33:1619, 33:1622,
 33:1623, 33:1624, 33:1629, 33:1633,
 01:228, 01:241, 06:3, 01:254,
 06:36, 06:43, 01:267, 46:8, 46:25,
 48:326, 48:327, 48:331, 48:335,
 48:339, 48:343, 48:347, 50:384,
 50:400, 50:412, 01:305, 50:1589,
 50:1796, 50:1799, 06:614, 06:615,
 53:104, 53:115, 53:148, 06:649,
 06:650, 06:655, 06:709, 06:710,
 06:725, 06:727, 54:1922, 54:1934,
 55:118, 57:276, 57:652, 57:659,
 57:671, 57:672, 57:680, 01:57, 103
\typeoutdetails 33:1216, 33:1217,
 33:1220, 33:1226, 33:1228, 33:1232,
 33:1233, 33:1239, 33:1257, 33:1258,
 33:1412, 33:1627, 33:1629, 33:1633

U
\u 21:237, 21:386, 21:472, 21:589,
 21:596, 21:616, 21:623, 21:754,
 21:1242, 21:1317, 21:1318, 21:1333,
 21:1334, 21:1343, 21:1344, 21:1357,
 21:1358, 21:1359, 21:1383, 21:1384,
 21:1409, 21:1410, 33:163, 33:194
\uccode 57:240, 57:248, 57:255,
 57:257, 57:260, 57:262, 57:535,
 57:543, 57:550, 57:552, 57:555, 57:557
\Ucharcat 29:624
\uchyph 02:370
\ulcdefault 25:526, 25:605
\ulcshape 25:526, 25:604,
 25:698, 25:699, 29:549, 32:29, 565
\Umathcode 04:30, 18:552, 30:15,
 30:51, 30:58, 30:73, 33:159, 33:367,
 37:577, 57:160, 57:351, 57:529, 02:127
\unboldmath 29:615
\UndeclareTextCommand 21:191,
 33:115, 33:117, 33:119, 33:308,
 33:585, 33:1127, 33:1128, 33:1129,
 33:1130, 33:1131, 33:1132, 1209
\undefined 22:119, 22:120,
 22:176, 22:177, 22:178, 22:179,
 22:180, 22:181, 01:12, 30:123,
 30:124, 01:14, 35:40, 35:41, 35:57,
 35:58, 35:59, 35:60, 01:20, 57:168,
 57:180, 57:181, 57:201, 01:41, 1229
\undefinedpagestyle 49:4, 49:8
\underbar 06:893, 06:914, 02:522
\underbrace 30:547
\underline 819
\underline 40:551, 40:552, 02:522
\underscore 1235
\unexpanded 32:47, 37:279, 37:315, 50:824,
 50:1624, 50:1626, 06:632, 06:643, 1259
\unhbox 1196
\unhcopy 41:345, 42:741, 42:797, 02:524
\unicodedataline
 04:143, 04:146, 04:160, 04:161, 04:162

```

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=ltterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

\UnicodeEncodingName . . . . .  
     . . . . . 21:981, 21:987, 21:1038,  
     21:1044, 21:1048, 21:1059, 21:1063,  
     21:1079, 21:1080, 33:376, 33:377,  
     33:378, 33:379, 33:380, 33:381,  
     33:382, 33:383, 33:384, 33:385,  
     33:386, 33:387, 33:388, 33:389,  
     33:390, 33:391, 33:392, 33:393, 500  
\UnicodeFontFile . . . . . 21:1036  
\UnicodeFontName . . . . . 21:1037  
\UnicodeFontTeXLigatures . 21:993, 21:1033  
\unicoderead . . . . . 04:143,  
     04:157, 04:158, 04:159, 04:160, 04:165  
uninstall . . . . . 04:999  
\unitlength . . . . . 40:74, 40:85,  
     40:94, 40:104, 42:5, 42:29, 42:30,  
     42:32, 42:34, 42:42, 42:43, 42:44,  
     42:45, 42:60, 42:63, 42:73, 42:74,  
     42:84, 42:85, 42:93, 42:94, 42:107,  
     42:108, 42:119, 42:164, 42:176,  
     42:241, 42:256, 42:316, 42:318,  
     42:332, 42:340, 42:342, 42:357,  
     42:375, 42:377, 42:392, 42:397,  
     42:399, 42:414, 42:417, 42:422,  
     42:479, 42:480, 42:507, 42:508,  
     42:536, 42:537, 42:611, 42:627,  
     42:647, 42:653, 42:689, 42:690,  
     42:692, 42:693, 42:696, 42:697,  
     42:699, 42:700, 42:711, 42:712,  
     42:714, 42:715, 42:717, 42:718,  
     42:720, 42:721, 42:749, 42:750,  
     42:752, 42:753, 42:756, 42:757,  
     42:759, 42:760, 42:771, 42:773,  
     42:775, 42:777, 53:328, 57:149, 1061  
\unkern . . . . . 24:701  
\nunless . . . . . 04:151, 04:159, 04:161  
\unlhd . . . . . 29:732  
\unpenalty . . . . . 24:704,  
     24:708, 32:116, 37:543, 37:565, 409  
\unrhd . . . . . 29:734  
\unsetattribute . . . . . 43  
\unsetattribute . . . . . 04:82, 04:240  
\unskip . . . . . 1196  
\unvcopy . . . . . 38:193, 949  
\Uparrow . . . . . 30:591  
\uparrowarrow . . . . . 30:585  
\upbracefill . . . . . 30:550, 30:567  
\updefault . . . . . 25:688, 29:21, 30:102,  
     30:109, 30:110, 30:118, 30:120, 668  
\Updownarrow . . . . . 30:595  
\updownarrow . . . . . 30:589  
\uplus . . . . . 30:385  
\uppercase . . . . . 1209  
\upshape . . . . . 21:442,  
     21:732, 21:802, 21:895, 25:686,  
     25:687, 29:19, 29:20, 29:549, 29:560,  
     29:593, 29:651, 32:24, 33:629, 565  
\Upsilon . . . . . 30:311  
\upsilon . . . . . 30:293  
use commands:  
\use:N . . . . . 08:2289, 08:2359, 08:2375, 10:136,  
     10:143, 11:273, 11:489, 11:630,  
     11:756, 11:905, 28:295, 28:334,  
     36:73, 05:181, 48:112, 48:129,  
     48:142, 48:160, 48:179, 48:186,  
     48:232, 52:442, 52:446, 52:448,  
     52:465, 57:591, 07:804, 07:1455,  
     08:1629, 08:1762, 08:1763, 08:1803, 339  
\use:n . . . . . 08:2104, 08:2341, 08:2347,  
     08:2356, 09:105, 09:172, 09:175,  
     09:203, 09:212, 09:247, 09:250,  
     09:278, 09:284, 09:350, 09:383,  
     09:406, 09:425, 09:454, 09:484,  
     09:506, 09:513, 09:549, 11:294,  
     11:338, 11:372, 20:523, 48:111,  
     51:180, 07:186, 07:683, 07:686,  
     07:862, 07:1095, 07:1222, 07:1314,  
     07:1423, 07:2350, 07:2392, 07:2965,  
     07:2967, 07:2972, 08:238, 08:388,  
     08:511, 08:1152, 08:1627, 08:2070, 955  
\use:nn . . . . . 08:2120,  
     08:2679, 09:217, 09:289, 08:1733  
\use:nnn . . . . . 07:2811, 07:2825, 07:2930, 07:2944, 188  
\use\_i:nn . . . 08:2206, 08:2551, 49:41,  
     07:946, 07:1634, 07:2445, 07:2498, 346  
\use\_i:nnn . . . . . 07:2423, 08:2069, 263  
\use\_i:nnnn . . . . . 09:348, 319  
\use\_i\_delimit\_by\_q\_recursion\_-  
     stop:nw . . . . . 09:413, 09:432  
\use\_i\_delimit\_by\_q\_stop:nw . 07:426  
\use\_ii:nn . . . . . 08:2557,  
     09:57, 09:139, 55:24, 55:34,  
     07:1536, 07:1556, 07:1561, 07:1566,  
     07:1571, 07:1816, 07:2025, 07:2038,  
     07:2077, 07:2095, 07:2120, 07:2153,  
     07:2180, 07:2444, 07:2976, 08:637, 169  
\use\_ii:nnn . . . . .  
     . . . 07:1617, 07:2094, 07:2427, 07:2969  
\use\_ii:nnnn . . . . . 09:353  
\use\_ii\_i:nn . . . . . 09:216, 09:288  
\use\_ii\_iii:nnn . . . . .  
     . . . 52:216, 52:230, 52:242, 52:242  
\use\_iii:mn . . . . . 263  
\use\_iii:nnn . . . . . 52:47, 07:2054,  
     07:2422, 07:2425, 07:2497, 07:2971

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

```

\use_iii:nnnn 07:2053 \UseOneTimeHook 08:2784, 08:2903, 20:15,
\use_iv:nnnn 07:412, 07:418 20:57, 20:70, 20:317, 20:321, 20:326,
\use_none:n 08:2208, 08:2297, 20:339, 20:378, 20:380, 20:383,
 09:355, 09:413, 09:432, 10:7, 10:8, 37:14, 37:18, 37:29, 37:30, 37:32,
 48:305, 52:131, 52:134, 53:7, 55:3, 37:80, 37:84, 37:94, 37:95, 37:97,
 55:4, 55:23, 55:33, 55:121, 55:123, 50:1004, 50:1008, 50:1030, 50:1034, 212
 07:95, 07:687, 07:1577, 07:1583, \UseOneTimeHookWithArguments 08:2784, 197
 07:1589, 07:1595, 07:1612, 07:1613, \usepackage ... 50:676, 50:738, 50:1604, 200
 07:1616, 07:1619, 07:1633, 07:1660, \UseRawInputEncoding 57:381, 57:437, 57:484
 07:1797, 07:2089, 07:2090, 07:2093, \UseSocket 10:170,
 07:2096, 07:2300, 08:7, 08:638, 10:188, 35:123, 35:128, 35:133, 330
 08:1348, 08:1419, 08:1466, 08:1731, 283 \UseTaggingSocket 1161
\use_none:nn 08:2102, 28:294, 28:333, 48:263, \UseTaggingSocket 17:40, 35:129,
 07:1112, 07:1791, 07:2048, 07:2049, 55:7, 55:173, 55:209, 55:224, 1162
 07:2052, 07:2055, 07:2329, 08:2061, 961 \UseTemplate 11:92, 11:1156, 354
\use_none:nnn ... 07:873, 07:1785, \UseTextAccent 21:149,
 07:2046, 07:2488, 07:2863, 07:2869, 21:150, 21:188, 33:110, 33:111,
 07:2880, 07:2886, 08:744, 08:2067, 170 33:113, 33:156, 33:158, 33:936,
\use_none:nnnn 08:2928, 07:1779, 07:2303 33:1121, 33:1122, 33:1124, 33:1125, 476
\use_none:nnnnn 07:1773, 08:1840, 08:1958 \UseTextSymbol 21:150, 21:186,
\use_none:nnnnnn 07:1767 33:109, 33:352, 33:935, 33:1004, 476
\use_none:nnnnnnn 07:1761 \usetikzlibrary 297
\use_none:nnnnnnnn 08:1348, 08:1840, 08:1958 \ushape 1236
\usebox 40:195
\usecounter 39:249, 39:262
\UseExpandableTaggingSocket 1161
\UseExpandableTaggingSocket . 55:7, 1161
\usefont 21:1574, 24:80, 24:282, 24:677,
 29:709, 33:7, 33:613, 37:581, 1228
\UseHook 08:2784,
 08:2902, 09:290, 09:557, 09:560, 21:238, 21:387,
 20:316, 20:322, 20:327, 20:338, 21:471, 21:592, 21:593, 21:594,
 20:377, 20:381, 20:384, 26:145, 21:598, 21:600, 21:603, 21:605,
 29:263, 29:274, 29:286, 29:296, 21:607, 21:613, 21:619, 21:620,
 29:317, 29:324, 29:337, 29:344, 21:621, 21:625, 21:627, 21:630,
 29:403, 29:408, 29:413, 29:418, 21:632, 21:634, 21:640, 21:755,
 29:663, 29:680, 37:248, 37:260, 21:1247, 21:1327, 21:1328, 21:1329,
 37:263, 37:351, 37:355, 37:368, 21:1330, 21:1339, 21:1340, 21:1373,
 37:371, 50:1003, 50:1007, 50:1031, 21:1374, 21:1379, 21:1380, 21:1391,
 50:1035, 52:170, 52:171, 52:174, 21:1392, 21:1399, 21:1400, 21:1403,
 52:175, 53:118, 53:170, 53:382, 226 21:1404, 21:1426, 21:1427, 21:1428,
\UseHookWithArguments 08:2784, 21:1429, 21:1430, 21:1431, 21:1432,
 09:219, 09:522, 09:529, 35:104, 196 21:1433, 21:1434, 21:1435, 21:1436,
\UseInstance . 11:478, 11:1072, 11:1156, 348 21:1445, 21:1446, 21:1447, 21:1448,
\UseLegacyTextSymbols ... 33:576, 33:799 21:1451, 21:1452, 33:165, 33:195
\UserName 78
\UserName 05:177, 05:195

```

**File Key:** 01=ltdirname.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefsns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspacedtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfnctcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=ltbypnem.dtx, 57=ltfinal.dtx

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \vbadness . . . . .                                                               | 53:212, 53:214,<br>53:263, 53:265, 54:2244, 02:319, 1076                                                                                                                                                                                                                                                                                                           | <b>W</b>                                                                                                                                                                                                                                                                                                                       |
| \vbox . . . . .                                                                   | 1075                                                                                                                                                                                                                                                                                                                                                               | \wedge . . . . . 30:373, 30:375                                                                                                                                                                                                                                                                                                |
| vbox commands:                                                                    |                                                                                                                                                                                                                                                                                                                                                                    | \whatsit . . . . . 04:194, 43                                                                                                                                                                                                                                                                                                  |
| \vbox_set:Nn . . . . .                                                            | 48:74                                                                                                                                                                                                                                                                                                                                                              | \widehat . . . . . 30:534                                                                                                                                                                                                                                                                                                      |
| \vbox_set_to_ht:Nnn . .                                                           | 53:215, 53:266                                                                                                                                                                                                                                                                                                                                                     | \widetilde . . . . . 30:533                                                                                                                                                                                                                                                                                                    |
| \vbox_to_zero:n . . . . .                                                         | 53:326                                                                                                                                                                                                                                                                                                                                                             | \widowpenalties . . . . . 02:106                                                                                                                                                                                                                                                                                               |
| \vbox_unpack:N . . . . .                                                          | 48:82, 48:95,<br>48:96, 48:385, 48:416, 53:226, 53:276                                                                                                                                                                                                                                                                                                             | \widowpenalty . . . . . 24:691, 02:326                                                                                                                                                                                                                                                                                         |
| \vdash . . . . .                                                                  | 30:411                                                                                                                                                                                                                                                                                                                                                             | \width . . . . . 40:30, 1216                                                                                                                                                                                                                                                                                                   |
| \vdots . . . . .                                                                  | 30:517                                                                                                                                                                                                                                                                                                                                                             | \wlog . . . . . 04:6, 04:7,<br>04:8, 04:54, 37:45, 37:110, 50:414,<br>57:60, 02:40, 57:705, 02:145, 02:239,<br>02:254, 02:284, 02:299, 01:84, 1206                                                                                                                                                                             |
| \vec . . . . .                                                                    | 30:531                                                                                                                                                                                                                                                                                                                                                             | \wp . . . . . 30:319                                                                                                                                                                                                                                                                                                           |
| \vector . . . . .                                                                 | 14:269, 42:233, 42:816, 42:833                                                                                                                                                                                                                                                                                                                                     | \wr . . . . . 30:389                                                                                                                                                                                                                                                                                                           |
| \vee . . . . .                                                                    | 30:374, 30:376                                                                                                                                                                                                                                                                                                                                                     | \write . . . . . 29                                                                                                                                                                                                                                                                                                            |
| \verb . . . . .                                                                   | 37:575, 37:591, 37:604, 37:618,<br>37:630, 37:636, 37:642, 37:655,<br>37:667, 37:672, 37:680, 37:686,<br>37:696, 37:701, 37:714, 37:716, 1198                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                |
| \verb* . . . . .                                                                  | 1266                                                                                                                                                                                                                                                                                                                                                               | <b>X</b>                                                                                                                                                                                                                                                                                                                       |
| \verbatim (env.) . . . . .                                                        | 37:569                                                                                                                                                                                                                                                                                                                                                             | \x . . . . . 03:96, 03:99, 24:333, 24:334, 57:330, 57:332                                                                                                                                                                                                                                                                      |
| \verbatim . . . . .                                                               | 37:569                                                                                                                                                                                                                                                                                                                                                             | \xdef . . . . . 1193                                                                                                                                                                                                                                                                                                           |
| \verbatim* (env.) . . . . .                                                       | 37:584                                                                                                                                                                                                                                                                                                                                                             | \XeTeXcharclass . . . . . 24:665, 57:39,<br>57:47, 57:54, 57:67, 57:73, 57:82, 57:89                                                                                                                                                                                                                                           |
| \verbvisiblespace . . . . .                                                       |                                                                                                                                                                                                                                                                                                                                                                    | \XeTeXcharclassCL . . . . . 57:173                                                                                                                                                                                                                                                                                             |
| . . . . . 37:575, 37:577, 37:591, 37:596,<br>37:606, 37:610, 37:620, 37:624, 1248 |                                                                                                                                                                                                                                                                                                                                                                    | \XeTeXcharclassCM . . . . . 57:177                                                                                                                                                                                                                                                                                             |
| \Vert . . . . .                                                                   | 30:578, 30:580                                                                                                                                                                                                                                                                                                                                                     | \XeTeXcharclassEX . . . . . 57:174                                                                                                                                                                                                                                                                                             |
| \vert . . . . .                                                                   | 30:583                                                                                                                                                                                                                                                                                                                                                             | \XeTeXcharclassID . . . . . 57:171                                                                                                                                                                                                                                                                                             |
| \vfil . . . . .                                                                   | 33:102, 33:105,<br>33:928, 33:931, 42:565, 42:577,<br>53:393, 53:405, 54:177, 54:196,<br>54:414, 54:461, 54:631, 54:690, 02:511                                                                                                                                                                                                                                    | \XeTeXcharclassIS . . . . . 57:175                                                                                                                                                                                                                                                                                             |
| \vfilneg . . . . .                                                                | 02:511                                                                                                                                                                                                                                                                                                                                                             | \XeTeXcharclassNS . . . . . 57:176                                                                                                                                                                                                                                                                                             |
| \vfuzz . . . . .                                                                  | 49:90, 49:97, 53:211,<br>53:213, 53:262, 53:264, 02:391, 1076                                                                                                                                                                                                                                                                                                      | \XeTeXcharclassOP . . . . . 57:172                                                                                                                                                                                                                                                                                             |
| \vglue . . . . .                                                                  | 02:501                                                                                                                                                                                                                                                                                                                                                             | \XeTeXcharglyph . . . . . 21:1034                                                                                                                                                                                                                                                                                              |
| \vline . . . . .                                                                  | 41:366                                                                                                                                                                                                                                                                                                                                                             | \XeTeXdashbreakstate . . . . . 57:273                                                                                                                                                                                                                                                                                          |
| \voffset . . . . .                                                                | 02:407                                                                                                                                                                                                                                                                                                                                                             | \XeTeXglyph . . . . . 21:1034                                                                                                                                                                                                                                                                                                  |
| \vphantom . . . . .                                                               | 21:497, 21:513, 38:75                                                                                                                                                                                                                                                                                                                                              | \XeTeXinterchartclasses . . . . . 57:167, 57:200                                                                                                                                                                                                                                                                               |
| \vrule . . . . .                                                                  | 18:528, 21:293, 21:295, 21:502,<br>21:518, 26:190, 30:565, 30:566,<br>30:568, 30:569, 40:204, 40:206,<br>40:282, 40:289, 40:550, 40:603,<br>40:610, 41:186, 41:219, 41:347,<br>41:366, 42:227, 42:299, 42:302,<br>42:324, 42:333, 42:350, 42:359,<br>42:383, 42:391, 42:406, 42:413,<br>42:564, 42:577, 42:725, 42:781,<br>54:1855, 54:2264, 54:2297, 02:505, 1268 | \XeTeXinterchartoks . . . . .<br>. . . . . 57:168, 57:182, 57:183,<br>57:184, 57:185, 57:186, 57:187,<br>57:188, 57:189, 57:190, 57:191,<br>57:192, 57:193, 57:194, 57:195,<br>57:196, 57:201, 57:206, 57:207,<br>57:208, 57:209, 57:210, 57:211,<br>57:212, 57:213, 57:214, 57:215,<br>57:216, 57:217, 57:218, 57:219, 57:220 |
| \vsizer . . . . .                                                                 | 1231                                                                                                                                                                                                                                                                                                                                                               | \XeTeXmathcode . . . . . 57:161, 57:530                                                                                                                                                                                                                                                                                        |
| \vskip . . . . .                                                                  | 1196                                                                                                                                                                                                                                                                                                                                                               | \XeTeXrevision . . . . . 57:41                                                                                                                                                                                                                                                                                                 |
| \vspace . . . . .                                                                 | 18:377, 18:447, 18:448, 18:449                                                                                                                                                                                                                                                                                                                                     | \XeTeXuseglyphmetrics . . . . . 57:270, 57:272                                                                                                                                                                                                                                                                                 |
| \vsplit . . . . .                                                                 | 54:384, 54:431, 54:2246, 957                                                                                                                                                                                                                                                                                                                                       | \XeTeXversion . . . . . 29:623, 57:41                                                                                                                                                                                                                                                                                          |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | \Xi . . . . . 30:308                                                                                                                                                                                                                                                                                                           |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | \xi . . . . . 30:288                                                                                                                                                                                                                                                                                                           |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | xpos . . . . . 36:243, 751                                                                                                                                                                                                                                                                                                     |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | \xpt . . . . . 1199                                                                                                                                                                                                                                                                                                            |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | \xspaceskip . . . . . 02:422                                                                                                                                                                                                                                                                                                   |
|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                    | \xtxHanGlue . . . . . 57:180, 57:204,<br>57:212, 57:213, 57:214, 57:215,<br>57:216, 57:217, 57:218, 57:219, 57:220                                                                                                                                                                                                             |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
 06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmhooks.dtx, 10=ltsockets.dtx,  
 11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
 16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspaced.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
 21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
 26=ltfsstrc.dtx, 27=ltfscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
 31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=ltxref.dtx,  
 36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
 41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
 46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
 51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
 56=lthyphen.dtx, 57=ltfinal.dtx

|                    |                                                                   |                 |                                  |
|--------------------|-------------------------------------------------------------------|-----------------|----------------------------------|
| \xtxHanSpace . . . | 57:181, 57:205, 57:206,<br>57:207, 57:208, 57:209, 57:210, 57:211 | ypos . . . . .  | 36:243, 751                      |
|                    | Y                                                                 |                 | Z                                |
| \y . . . . .       | 03:97, 03:99                                                      | \z . . . . .    | 57:252, 57:526, 57:547           |
| \year . . . . .    | 03:14, 01:169,<br>50:1298, 50:1431, 50:1520, 02:386               | \zeta . . . . . | 57:243, 57:527, 57:538<br>30:280 |

**File Key:** 01=ltdirchk.dtx, 02=lpplain.dtx, 03=ltvers.dtx, 04=ltluatex.dtx, 05=ltexpl.dtx,  
06=ltdefns.dtx, 07=ltcmd.dtx, 08=lthooks.dtx, 09=ltcmdhooks.dtx, 10=ltsockets.dtx,  
11=lttemplates.dtx, 12=ltalloc.dtx, 13=ltcntrl.dtx, 14=lterror.dtx, 15=ltpar.dtx,  
16=ltpara.dtx, 17=ltmeta.dtx, 18=ltspare.dtx, 19=ltlogos.dtx, 20=ltfiles.dtx,  
21=ltoutenc.dtx, 22=ltcounts.dtx, 23=ltlength.dtx, 24=ltfssbas.dtx, 25=ltfssaxes.dtx,  
26=ltfsstrc.dtx, 27=ltfsscmp.dtx, 28=ltfssdcl.dtx, 29=ltfssini.dtx, 30=fontdef.dtx,  
31=preload.dtx, 32=ltfntcmd.dtx, 33=lttextcomp.dtx, 34=ltpageno.dtx, 35=lxref.dtx,  
36=ltproperties.dtx, 37=ltmisen.dtx, 38=ltmath.dtx, 39=ltlists.dtx, 40=ltboxes.dtx,  
41=lttab.dtx, 42=lpictur.dtx, 43=ltthm.dtx, 44=ltsect.dtx, 45=ltfloat.dtx,  
46=ltidxglo.dtx, 47=ltbibl.dtx, 48=ltmarks.dtx, 49=ltpage.dtx, 50=ltclass.dtx,  
51=ltkeys.dtx, 52=ltfilehook.dtx, 53=ltshipout.dtx, 54=ltoutput.dtx, 55=lttagging.dtx,  
56=lthyphen.dtx, 57=ltfinal.dtx