The file `phddoc.dtx` for use with LaTeX 2_ε.* It contains the code for `phddoc.cls`

Yannis Lazarides

2005/11/02

*This file has version number v4.41, dated 2005/11/02.

STNETNOC FO ELBAT

@tocrmarg=4em
@pnumwidth=1.5em
section
indent=1.5em
subsection
indent=3.8em
subsubsection
indent=7.0em

Contents 1

1 The LateX Kernel 30

1.1 Idiomatic LaTeX programming 31

1.2 Code Organization. 33

1.3 Autoloading 34

1.4 File a ltdirchk.dtx 34

2 Plain TeX 36

2.1 Allocation of registers 38

2.2 Macros for setting ordinary text 47

2.3 Definitions related to output 54

3 Version Identification 57

4 LaTeXe Kernel Definitions Module 61

4.1 Introduction 62
4.2 Saved versions of TEX primitives 62
4.2.1 Command definitions 64
4.2.2 \@cons and \@car 65
4.3 Code 65
4.3.1 csname abstractions 65
4.4 List helper macros 66
4.5 Command Building 68
4.5.1 Checking if a command is definable 71
4.6 Environment Building commands 72
4.7 newif 73
4.8 Provide version of commands 74
4.8.1 Argument gobbling command utilities 76
4.8.2 Other argument grabbers 76
4.9 Robust commands and protect 77
4.9.1 Internal defining commands 80
4.9.2 Commands for Autoloading 83

5 Allocations 84

5.1 Counters 85
5.2 Scratch registers 85

6 Control Structures 87

6.1 Introduction 88

7 Error Handling (file lterror.dtx) 91

7.1 Error handling 92
7.1.1 General commands 92
7.1.2 Package and Class error messages 95
7.1.3 Specific errors 98

8 ltspace.dtx 103

8.1 Spacing 104
8.1.1 User Commands 104
8.1.2 Vertical spacing commands 105
8.1.3 Horizontal spacing 106
8.1.4 Obsolete Commands 106
8.1.5 The code 107
8.1.6 Space removing macros 109
8.1.7 Vertical spacing 111
8.1.8 Horizontal space (and breaks) 115

9 Logos 118

10 File Handling 120

10.1 Variables, switches and internal commands 121
10.2 Safe Input Macros 128
10.3 Listing files 130

11 ltoutenc 132

11.1 Background 133

11.2 Output encodings 133

11.2.1 Output encoding files 134

11.3 Font encodings 135

11.3.1 Removing encoding-specific commands 138

11.3.2 The order of declarations 138

11.3.3 Docstrip modules 139

11.3.4 Definitions for the kernel 139

11.3.5 Default encodings 147

11.3.6 Definitions for the OT1 encoding 150

11.3.7 Definitions for the T1 encoding 151

11.3.8 Definitions for the OML encoding 156

11.3.9 Definitions for the OT4 encoding 156

11.3.10 Definitions for the TS1 encoding 158

11.3.11 The textcomp package 163

12 Counters and Lengths 173

12.1 Counters and Lengths 174

12.1.1 Environment Counter Macros 174

13 Kernel Lengths 180

13.1 File n, lengths and the ltlength.dtx 181

13.2 Redefinitions and Extensions by Packages 183

14 The new font selection scheme for TEX macro packages 184

14.1 Preliminary macros 186

14.2 Macros for setting up the tables 186

14.3 Macros for loading fonts 195

15 ltfsstrc 204

15.1 A driver for this document 205

15.2 The Implementation 205

15.3 Handling Options 205

15.3.1 General font loading 207

16 Setting up Fonts 218

16.1 Introduction 219

16.2 The Choice of Typesetting Engine 219

16.3 What is a character? And a glyph? 221

16.4 What's a font? 221

16.5 The low-level interface 222
16.5.1 Encoding Schemes 222

16.6 Setting several font attributes 224
16.6.1 The Postscript fonts 225
16.6.2 Additional free fonts for use with LaTeX 229
16.6.3 Uncial fonts 229

16.7 Dingbat and Symbol Fonts 230
16.7.1 Unicode Dingbats block 231

16.8 Naming digital fonts 232
16.8.1 Karl Berry naming scheme 232

16.9 The phd package interface. 233

16.10 Viewing and selecting fonts 234
16.10.1 Typefaces that come with the standard LaTeX distribution 234

16.11 Discussion 234

16.12 XeLaTeX and LuaLaTeX 234

16.13 Utilities for testing fonts 235

16.14 `.tfm` files 236

16.15 Fonts for Far East Languages 236
16.15.1 Selecting a font 236

16.16 Entering CJK text 236

16.17 Changing the font size in LaTeX 237
16.17.1 Changing the font size on the document-wide level 237

16.18 Changing the font size locally 237

17 ltfssini 238

17.1 NFSS Initialisation 240
17.1.1 Providing math *versions* 240

18 Kernel fontdef.dtx 245

18.1 Introduction 246
18.2 The `fonttext.ltx` file 246
18.3 Math font sizes 248
18.3.1 The math symbol assignments 248
18.3.2 Delimitercodes for characters 250
18.3.3 Symbols accessed via control sequences 250
18.3.4 Greek letters 250
18.3.5 Ordinary symbols 251
18.3.6 Large Operators 252
18.3.7 Math versions of text commands 258
18.3.8 Other special functions and parameters 258
18.3.9 The log-like functions 258
18.3.10 Parameters 259
18.4 Default cfg files 259

19 Preload 260

19.1 Introduction 261
19.2 Overview 261
19.3 Customization 261
19.4 Module switches for the DOCSTRIP program 262
19.5 A driver for this document 262
19.6 The code 262

20 The ltfntcmd module 265

20.1 Introduction 266
20.2 The implementation 268

21 Page Number 272

21.1 Page Numbering 273

21.2 Modifications and additions by packages 273

22 ltxref.dtx 274

22.1 Cross Referencing 275

22.2 Author commands 275

22.2.1 Cross Referencing 275

22.2.2 An extension of counter referencing 278

23 Miscellaneous class 280

23.1 Environments 281

23.2 Center, Flushright, Flushleft 284

23.3 Verbatim 286

24 Kernel Maths 289

24.1 Math commands based on plain TEX 290

24.1.1 The log-like functions 290

24.1.2 Delimiters Biggggg 291

25 ltlists.dtx 304

25.1 List, and related environments 305

25.2 Generic Command 305

25.2.1 List and Trivlist 308

25.2.2 Horizontal Spacing (dimens) 310

25.2.3 Default Values 310

25.3 The \list environment 317

25.4 Typesetting the label 323

25.5 Enumerate 326

25.6 The itemize environment 327

26 LaTeX Box commands 330

26.1 Typesetting boxed content 331

26.2 Saving material in a box 334

26.2.1 Some low-level constructs 343

26.3 Concluding remarks 344

27 Tabbing, Tabular and Array Environments 345

27.0.1 tabbing 346

27.1 array and tabular environments 349

28 The picture Environment 355

28.1 Introduction 356
28.2 The Basic Commands 356
28.3 Text and Formulae 357
28.4 multiput and linethickness 358
28.5 Some examples 359
28.6 picture package 360
28.7 pict2e 361
28.8 xpicture 361

29 Picture Mode 362

29.1 Circles and ovals 370
29.2 Curves 376

30 Theorem Environments 378

30.1 AmS Improvements 383

31 Sectioning Commands 384

31.1 Introduction 385
31.2 The Title 385
31.3 Sectioning 386
31.3.1 Initializations 390
31.4 Table of Contents etc. 391
31.4.1 Convention 391
31.4.2 Commands 391

32 ltfloat.dtx 395

32.1 Float types 396
32.1.1 Floating Environments 396
32.2 Marginal Notes 406
32.3 Footnotes 409

33 Indices 414

33.1 Preparing an index 417
33.1.1 Subindices 418
33.2 Multi-page Indexing 418
33.3 Summary of commands 418
33.4 Indexing Class Documentation 418

34 The makeindex program 419

34.1 Output format 422
34.2 Processing 423
34.3 Customization 424
34.4 Writing custom indexing commands 425
34.5 Multiple indices 425
34.6 How to I add index to the table of contents? 425
34.7 Limitations of the MakeIndex index processor 425
34.8 The xindy index processor 426
34.8.1 Internationalization 426

35 Index and Glossary Generation 427

36 Bibliography Kernel Class 430

36.1 Introduction 431

36.2 Bibliography Generation 431

36.2.1 Default definitions 434

37 LaTeX kernel page styles and related commands 435

37.1 Introduction 436

37.2 Page Style Commands 436

37.3 How a page style makes running heads and feet 436

37.4 marking conventions 436

38 Page Breaking and the Output Routine (OR) 441

In this chapter we discuss one of the most mysterious aspects of TEX the output routine. 442

38.1 Page breaking algorithm 442
38.1.1 Typographical constraints 443
38.2 The current page and the recent contributions list 443
38.3 When is the page builder activated? 444
38.4 The depth of the current page 444
38.5 The height of a box of text 445
38.6 The height of `\box255` 445
38.6.1 Dead cycles. 445
38.6.2 TEX's Page Number. 446
38.6.3 The `\vsplit` operation. 446
38.7 Communicating with the OTR: Marks 448
38.8 Insertions 449
38.9 \shipout 449
38.10 LATEX output routines 452
38.11 Packages 452
38.12 hooking at shipout 453
38.13 How to place a background image 455

39 ltoutput.dtx 456

39.1 Introduction 457
39.2 Floats 457
39.3 Page Layout Parameters 458
39.4 Page Style Parameters 459
39.5 Float placement parameters 460
39.6 Output Routine Variables 461
39.7 Calling the output routine 461
39.8 Functions used in the output routine 462
39.9 User commands that call affect the output routine 463
39.10 Float-handling mechanisms 464
39.11 Macros and data structures for processing floats 465
39.12 Box lists for float-placement algorithms 466
39.13 Float-Placement algorithms 467
39.14 Definition of Float Boxes (inserts) 468
39.15 Page layout parameters 469
39.16 Internal registers 471
39.17 The \output routine 475
39.17.1 Kludgeins 498
39.18 Float control 500
39.18.1 Floats on a text page 509
39.18.2 Floats on their own page or column 510

40 How to Develop your Own Class or Package 512

40.1 Introduction 513
40.2 What is a class? 513
40.3 Identifying your class 513
40.4 Class Options 514
40.5 Set-up the font sizes 515
40.5.1 Setting up the normalsize 515
40.6 Adjusting paragraph parameters 516
40.7 Formatting Chapters and Sections 517
40.8 Adjusting the Index 518
40.9 Provide some hooks 518
40.10 The final act of kindness to your users 519

41 How to Package Your Class 519

41.0.1 The .ins file 521
41.0.2 Generating messages 523
41.1 What to put in a .dtx file 524
41.1.1 User documentation 527
41.2 General tips for defining a Class 527
41.3 Declaring Options 528
41.4 Executing Options 528
41.5 Special Commands for class files 529
41.6 A minimal class 529

42 ltclass 530

42.1 Introduction 531

42.2 User interface 531
42.2.1 Option processing 532

42.3 Class and Package interface 532
42.3.1 Class name and version 532
42.3.2 Package name and version 533
42.3.3 Requiring other packages 533
42.3.4 Declaring new options 534
42.3.5 Safe Input Macros 535

42.4 Implementation 535

42.5 Passing Options 539

42.6 Declaring Options 540
42.6.1 Hooks 547
42.6.2 Providing shipment 549

42.7 After Preamble 552

43 lthyphen 553

44 Final settings 556

44.0.1 Debugging 557
44.0.2 Typesetting parameters 557
44.0.3 Lccodes for hyphenation 559

44.1 Hyphenation 563
44.1.1 Font loading 563
44.1.2 Input encoding 564
44.1.3 Lccodes and uccodes 565
44.1.4 Initialise file list 567
44.1.5 Dumping the format 567

45 LuaTeX support (file ltluatex) 568

45.1 Overview 569

45.2 Core TEX functionality 569

45.3 Plain TEX interface 570

45.4 Lua functionality 570

45.4.1 Allocators in Lua 570

45.4.2 Lua access to TEX register numbers 571

45.4.3 Module utilities 572

45.4.4 Callback management 572

45.4.5 Handlers 584

45.4.6 Public functions for callback management 586

I Latex Classes 591

46 The book.cls 593

46.1 General 595
46.2 Running Heads 598
46.3 Lists 607
46.4 The description Environment 608
46.5 Verse Environment 608
46.6 Quotation and quote environments 609
46.7 FIGURE AND TABLE ENVIRONMENTS 612
46.8 Defining the old style font commands 614
46.9 Table of contents 614
46.9.1 List of figures, tables etc 616
46.10 Bibliographies 617
46.11 The Index Environment 618
46.12 Footnotes 618
46.13 Catering for Other Languages 619
46.14 Final 619
46.15 Ending remarks 620
46.16 The .clo files 621

47 The classes module 626

47.1 A driver for this document 627
47.2 The DOCSTRIP modules 627
47.3 Initial Code 627
47.4 Declaration of Options 628
47.4.1 Setting Paper Sizes 628
47.4.2 Open bibliography 629
47.4.3 Page Layout 633
47.4.4 Lists 646

II Book Design 657

48 Creating Book Designs 659

48.1 First Steps 660
48.1.1 Chapter Opening 660
48.1.2 The User Commands 661
48.2 Sections 662
48.3 Examples and Solutions 663
48.4 Exercises 663
48.5 Figures and diagrams 663
48.6 Geometry 666

III The Ecosystem 667

49 Key Value Interfaces 669

49.1 keyval 670
49.2 xkeyval 670
49.3 Ordinary Keys 671
49.4 Keys and values in package options 671
49.5 kvoptions 672

50 Managing Keys with PGF 673

50.1 PGF keys 674

50.2 Introduction 675

50.2.1 Comparison to Other Packages 675

50.3 Technical 676

50.3.1 Getting started 677

50.3.2 The Key Tree 679

50.3.3 Setting Keys 680

50.3.4 Keys That Execute Commands 682

50.3.5 Keys That Store Values 684

50.3.6 Keys That Are Handled 684

50.3.7 Keys That Are Unknown 685

50.3.8 Key Handlers 686

50.3.9 Handlers for Path Management 686

50.3.10 Setting Defaults 687

50.3.11 Defining Key Codes 688

50.3.12 Defining Styles 690

50.3.13 Error Keys 698

IV expl3 699

A phddoc User Manual 701

A.1 Documentation of the LaTeX sources 703

A.2 Specification 703

A.3 Customisation 704

B Implementation Code 705

B.1 Options 707
B.1.1 Geometry parameters 707
B.2 Configuration 708
B.3 Option Processing 709
B.4 Loading book and doc 709
B.5 Useful abbreviations 710
B.6 DocInclude 711

Index 714

C phd-documentation Package, Code Implementation Objectives and Strategy 768

C.1 Requirements Specification 770

C.2 Terminology 771

C.3 Users 771
C.3.1 Author 771
C.3.2 Template designer 772
C.3.3 The programmer 772

C.4 Acknowledgements 772

C.5 Preliminaries 772

C.6 tcolorbox 773

C.7 Verbatim fonts 773

C.8 Keys dealing with colors 774
C.8.1 Go language 789

C.9 LaTeX code demo environments 790

C.10 makeidx 795

C.11 Documenting Macros 795
C.11.1 Keys for language related texts. 797

C.12 Index settings 797
C.12.1 Keys for adjusting design of command displays. 799

C.13 Indexing 810
C.13.1 Index heading and prologue 811

C.14 Miscellaneous doc commands 813

C.15 Other Indexing functions 813

C.16 Indexing Symbols 814
Table C.1: **textcomp** Currency Symbols 814
C.16.1 Indexing archaic scripts. 816
C.16.2 Indexing mathematical symbols 818
Table C.3: LaTeX Escapable "Special" Characters 822
Table C.4: **harmony** Musical Accents 829
Table C.6: AMS Commands Defined to Work in Both Math and Text Mode 832
Table C.7: Fontawsome Currency Symbols 837
846

C.17 Unicode math index functions 849

D l3 Documentation 851

D.0.1 General text markup 853
D.0.2 Describing functions in the documentation 854

D.1 Describing functions in the implementation 855

D.2 Variables 857
D.2.1 Variants and helpers 863
D.2.2 Messages 870
D.2.3 Configuration and tweaks 871
D.2.4 Implementing text markup 875
D.2.5 Inheriting doc 901
D.2.6 At end document 914
D.2.7 Indexing 916
D.2.8 Change history 922

E Code Implementation and Objectives 925

E.1 Terminology 927

E.2 Users 927
E.2.1 Author 927
E.2.2 Template designer 928
E.2.3 The Programmer 928

E.3 Color Palettes 928

E.4 Companion Go packages 928
E.4.1 Subsection heading example 929

E.5 Preliminaries 929

E.6 Palette and Color Naming Conventions 929
E.6.1 Material Design 930

E.7 Creating new palettes 931
E.7.1 Green Palettes 934

E.8 Hyperlinks 935

E.9 Code listings and documentation macros 935
E.9.1 Specific code coloring for l3 936

E.10 User Commands 936

F Handlers Package Code Implementation Objectives and Strategy 937

F.1 Terminology 939

F.2 Users 940
F.2.1 Author 940
F.2.2 Template designer 940
F.2.3 The programmer 940

F.3 Preliminaries 940

F.4 Source2e Interface 941

F.5 PGF Key Handlers 941
F.5.1 Font family handler 941
F.5.2 Font face handler 942
F.5.3 Font face handler 943
F.5.4 Fonts style handler 943
F.5.5 Capitalization handler 946
F.5.6 Alignment handler 947
F.5.7 On off general handlers 948
F.5.8 Shadow handlers 949
F.5.9 Sectioning format handler 949
F.5.10 Number format handler 951

V IMPLEMENTATION AND FRIENDS

954

G phd-sect Code Implementation Objectives and Strategy 956

G.1 Terminology 957

G.2 Users 958
G.2.1 Author 958
G.2.2 Template designer 958
G.2.3 The programmer 958

G.3 Preliminaries 958

G.4 Source2e Interface 959

G.5 Key Management 960
G.5.1 Inheritance, auto and cascade 969
G.5.2 Creating the Keys 969

G.6 Programmatically create key defaults 970

H Formatters and Associated Functions 972

H.1 Helper functions 973

H.2 Hanged formats 974

H.3 Format block 976
H.3.1 Display format 980

I Layout Engine Code 982

I.0.1 Paragraphs 994
I.0.2 Subparagraph 995

J Default Settings 996

J.1 Styles 1003

K Tests 1004

L The phd-i18n Package 2

L.1 Usage 3

A Implementation 4

A.1 Specification 5
A.2 Data 6
A.3 Preliminaries 6
A.4 Keys 6
A.4.1 Numbers 7
A.5 Retrieving the translation strings 9
A.6 Functions generated via scripts 10
A.7 Asturian 10
A.8 Amharic 11
A.9 Greek 12
A.10 German 13
A.11 French 14
A.11.1 French months 14
A.12 Italian 16
A.13 Albanian 17
A.14 Malayalam 18
A.15 Russian 19
A.16 Basque 20
A.17 Brazil 21
A.18 Breton 22
A.19 Bulgarian 23
A.20 Catalan 24
A.21 Croatian 25
A.22 Czech 26
A.23 Danish 27
A.24 Estonian 28
A.25 Finnish 29
A.25.1 Lists 30
A.26 Friulan 32
A.27 Galician 34
A.28 Galician 34
A.29 Icelandic 35
A.30 Irish 36
A.31 Latin 36
A.32 Latvian 37
A.33 Lithuanian 38
A.34 Lsorbian 39
A.35 Magyar (Hungarian) 39

Index 66

h a n g

1

The LateX Kernel

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

He who would eat the kernel, must crack the shell. [Lat., *Qui e nuce nucleum esse vult, frangat nucem.*]

Plautus

LaTeX is the batteries of TeX. It has given authors the ability to write documents easily and in a consistent way. It has also provided pacakage writers with endless nights of work trying to figure out how things work. In this chapter we will describe briefly the workings of LaTeX and the areas where one could use it for improvements. The best source for how the LaTeX kernel works is LaTeX itself and the publication that comes with it `sourc2e.pdf`.

The original developer of LaTeX was Leslie Lamport. Lamport who is now a recipient of the Turing award, described his philosophy for document production in a paper *Document Production: Visual or Logical?*, presented before the ACM in 1987. (**lamport1987**)

Although many developers have criticized the original code it is an amazing fact that it has endured. Although in today's terms the system cannot be considered pluggable, it was certainly extensible and the fact that TeX provides the means to write macros on the fly, the main effort of the LaTeX system was to provide the basic structure for the development of styles (now called classes).

The code was and is well documented and many of the comments are from the first system. I have spent many an evening thinking that analyzing latex code, should be mandatory in modern computer classes to instill good discipline in being economical with code and for discovering the right amount of abstraction before designing programming and delving into code.

1.1 Idiomatic LaTeX programming

My primary reason for recasting the lisitng of the LaTeX kernel code was educational. Programming in any language has its idioms and the only way to learn it is by reading other people's programs. Also I am not aware of any book that deals with LaTeX as a programming framework rather than a typesetting framework.

Of course the ultimate reason why LaTeX exists is to typeset documents. The kernel provides the structure and the necessary macros to enable the definition of the LaTeX classes such as books.

Most of the write-up is straight from the kernel files. Now and then I have added a bit of extra explanations, so the credit belongs to the LaTeX Team and to Leslie Lamport. Any mistakes are mine.

Variable and function names

1. First and foremost use of `@` to avoid conflict between user names and internal control sequences. As a prefix: `\@latexerror`. As a suffix to mark auxiliary macros such as `\newmathalphabet@@` or `\newmathalphabet@@@`. When used as a suffix it mostly avoids prefixes. To separate the words in a variable `\scan@fontshape`, `\end@float`.

2. Variables and macro names are mostly in lowercase. If they do not have

a @, expected to be used by authors or package writers. \topmargin, \textheight. Occassionaly uper and lower cased.

3. If a macro name had o use @ \hb@xt@.
4. Usage of macros for common numbers \@ne, \@cclv, \z@.
5. Usage of scatch macros and registers. This caused me a lot of grief when I was learning. \@tempa, \@tempboxa.
6. Usage of \@@ as a prefix to TEX primitives, \@@par, \@@input, \@@end.

There is no consistency in the convention, which is understable, so the above are just indicative.

Errors

1. Error trapping, by checking if a command is defined, allowed to be defined etc, before defining it. This partially overcomes the namespacing problems, at least until the point of definitions.

Use of lists

1. Use of lists extensively. There are three types of lists used. The first one are straight from the plain TeX code, whereas the second and third types were developed by the LATEX Team. Some go to the initial code by Lamport.

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

   Comma delimited lists are used in numerous places. These are normally constructed using special commands. For example:

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

   Iterating through the list can get complicated but one can use a \@for loop.

User interface

1. The kernel tries to provide user commands that have semantic meaning. An extreme example is \install@mathalphabet where it is let to \gdef.

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

Frugality

1. The use of scratch variables for example \@tempboxa is an amazing example of what I call code thriftiness. Firstly there is only one scratch box register ad is used everywhere from the float and output routines to the picture environment. Scary sometimes, that a small mishap somwhere can case chaos to a far off land.
2. Undefining variable sthat are no longer required \@onlypreamble type of commands.

Backward compatibility

Stability

1.2 Code Organization.

The LaTeX source code is distributed in a number of `.dtx` files. These are saved in files `a`..`z` and files `A-O` The source files are documented in `source2e`, just `texdoc source2e` to read it.

What I am describing here, is a step by step analysis of the kernel code, supplemented by additional materials, in order to provide an understanding of the inner workings.

	Filename	Description	Reference
a	ltxdirchk.dtx		
b	ltplain.dtx	Definitions from PlainTeX	Chapter **??**, page **??**
c	ltvers.dtx	Version information	Chapter **??**, page **??**
d	ltdefns.dtx	Definitons	Chapter **??**,pg **??**
e	ltalloc.dtx	Allocations	Chapter **??**, page **??**
f	ltcntrl.dtx	Control structures	Chapter **??**, page **??**
g	lterror.dtx	Error handling	Chapter **??**, page **??**
h	ltpar.dtx	Paragraph related	Chapter **??**, page **??**
i	ltspace.dtx	Spacing commands	Chapter **??**, page **??**
j	ltlogos.dtx	Logos	Chapter **??**, page **??**
k	files.dtx	File handling	Chapter **??**, page **??**
l	ltoutenc.dtx	Output encodings	Chapter **??**, page **??**
m	ltcounts.dtx	Counters	Chapter **??**, page **??**
n	ltlengths	Length macros	Chapter **??**, page [**??**]
o	ltfssbas.dtx	Low level font commands	Chapter **??**,page **??**
p	ltfsstrc.dtx	Tracing fonts	Chapter **??**, page **??**
q	ltfsscmp.dtx		Chapter **??**, page **??**
s	ltfssini.dtx	NFSS font initialization	Chapter **??**, page **??**
t	fontdef.dtx	font definitions	Chapter **??**, page **??**
u	preload.dtx	font preloading	Chapter **??** page **??**
v	ltfntcmd.dtx	font commands	**??**
w	ltpageno.dtx	page numbers	Chapter **??**, page [**??**]
x	ltref.dtx	References	Chapter **??**, page **??**
y	ltmiscen.dtx	Miscellaneous	Chapter **??**, page **??**
z	ltmath.dtx	Math	Chapter **??**, page **??**
A	ltlists.dtx	Lists	Chapter **??**, page **??**
B	ltboxes.dtx	Boxes	Chapter **??**, page **??**
C	lttab.dtx	Tabular and tabbing	Chapter **??**, page **??**
D	ltpictur.dtx	Picture	Chapter **??**, page **??**
E	ltthm.dtx	Theorems	Chapter **??**, page **??**
F	ltsect.dtx	Sectioning	Chapter **??**, page **??**
G	ltfloat.dtx	Floats	Chapter **??**, page **??**
H	ltidxglo.dtx	Indices & Glossaries	Chapter **??**, page **??**
I	ltbibl.dtx	Bibliography	Chapter **??**, page **??**

J	ltpage.dtx	Page	Chapter**??**, page [**??**]
K	ltoutput.dtx	Output routine	Chapter **??**, page **??**
L	ltclass.dtx	Class and package definitions	Chapter **??**, page [**??**]
M	lthyphen.dtx	Hyphenation	Chapter **??**, page [**??**]
N	ltfinal.dtx	Final	Chapter **??**, page **??**
O	ltluatex	LuaTeX support	Chapter **??**, page [**??**]

1.3 Autoloading

When LaTeX2e was released, personal computers had much less power than nowadays. Moreover, TeX was often compiled with a rather small amount of available memory. [1].

The inclusion of the New Font Selection Scheme (NFSS2), in particular, posed some challenges when LaTeXing big documents. So the developers provided a solution for people with limited memory available: if the autoload option was set during the extraction of latex.ltx from the sources, not all the kernel was included in the format which was then produced by running initex on this file: some parts of it were included "on demand", for instance the code for the picture environment.

Support of the autoload feature was introduced in the June 1995 release of the LaTeX kernel update and dropped in December 2003.

You can still find a description of this feature in the file

```
<TEX DIST ROOT>/doc/latex/base/autoload.txt
```

On some small systems (perhaps most noticeably emTeX for PCs if your machine is unable to use the TeX386 version) LaTeX uses up a large amount of the memory available to TeX, leaving very little for storage of any further commands, complex text (such as tables), floats or cross references that may occur in a typical document. Note that these limits are built into the TeX executable and do not directly correspond to any physical memory that your machine has installed.

In order to help with this problem, we have produced an experimental configuration of LaTeX in which certain functions are not predefined in the format, but are loaded automatically from a style file the first time they are used. This saves a lot of memory in the case that a document does not use these features.

In this release two environments are 'auto-loaded' in this way, 'picture' and 'tabbing', as are various bits of internal code used in error handling, font loading and advanced page makeup.

autopict.sty source for picture mode autotabg.sty source for tabbing environment autoerr.sty texts of most LaTeX error commands autofss1.sty little used internal font selection commands autoout1.sty source for `\enlargethispage` and related commands.

[1] http://tex.stackexchange.com/questions/38436/what-is-autoload

1.4 File a ltdirchk.dtx

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

`\input@path` For most common operating sytsems is let to undefined.

The routines define a useful macro to parse file name paths:

Example 4.1 Parsing directories

```
\filename@parse{./test/some other paths/path/tex.jpg}

\filename@area

\filename@base,

\filename@ext
```

./test/some other paths/path/
tex,
jpg

The `\@TeXversion` is only defined for very old versions of `TeX`, on a reasonable modern distribution, it should be let to undefined.

h a n g

2

Plain TeX

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

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

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

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

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

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

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

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

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

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

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

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

```
\catcode`@=11
```

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

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

Small constants are defined using `\chardef`.

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

(*End definition for* `\@ne` *and others.*)

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

Constants above 255 defined using **\mathchardef**.

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

(*End definition for* `\@cclvi` *and others.*)

2.1 Allocation of registers

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

```
\message{registers,}
```

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

New counters are allocated starting with 23, 24, etc. Other registers are allocated starting with 10. This leaves 0 through 9 for the user to play with safely, except that counts 0 to 9 are considered to be the page and subpage numbers (since they are displayed during output). In this scheme, **\count** 10 always contains the number of the highest-numbered counter that has been allocated, **\count** 14 the highest-numbered box, etc. Inserts are given numbers 254, 253, etc., since they require a **\count**, **\dimen**, **\skip**, and **\box** all with the same number; **\count** 20 contains the lowest-numbered insert that has been allocated. Of course, **\box**255 is reserved for **\output**; **\count**255, **\dimen**255, and **\skip**255 can be used freely.

It is recommended that macro designers always use **\global** assignments with respect to registers numbered
1, 3, 5, 7, 9,
and always non-**\global** assignments with respect to registers
0, 2, 4, 6, 8, 255.
This will prevent "save stack buildup" that might otherwise occur.

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

\insc@unt
\allocationnumber

The insertion counter and most recent allocation.

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

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

`\m@ne`

The constant −1.

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

`\wlog` Write on log file (only)

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

(*End definition for* `\wlog`.)

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

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

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

(*End definition for* `\count@` *and others*.)

`\newdimen` `\newskip` `\newmuskip` `\newbox` `\newread` `\newwrite` `\newlanguage`

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

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

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

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                    {\newcount}{Extended ↩
      ↪Allocation}%
\def\newcount {\e@alloc\count \countdef ↩
      ↪{\count10}\insc@unt\float@count}
\def\newdimen {\e@alloc\dimen \dimendef ↩
      ↪{\count11}\insc@unt\float@count}
\def\newskip  {\e@alloc\skip  \skipdef  ↩
      ↪{\count12}\insc@unt\float@count}
\def\newmuskip
           {\e@alloc\muskip\muskipdef{\count13}\m@ne\e@alloc@top}
\def\newbox   {\e@alloc\box
                  {\ifnum\allocationnumber<\@cclvi
                     \expandafter\chardef
                   \else
                     \expandafter\e@alloc@chardef
                   \fi}
                                        {\count14}\insc@unt\float@count}
\def\newtoks  {\e@alloc\toks ↩
      ↪\toksdef{\count15}\m@ne\e@alloc@top}
\def\newread  {\e@alloc\read \chardef{\count16}\m@ne\sixt@@n}
```

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

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

```
\ifx\directlua\@undefined
  \def\newlanguage  {\e@alloc\language ⤸
        ⤹\chardef{\count19}\m@ne\@cclvi}
\else
  \def\newlanguage  {\e@alloc\language ⤸
        ⤹\chardef{\count19}\m@ne{16384}}
\fi
%</2ekernel|latexrelease>
```

`\newcount{cmd}`

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

(*End definition for* `\newdimen` *and others.*)

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

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

```
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\e@alloc@chardef}{Extended ⤸
      ⤹Allocation}%
```

```
\ifx\directlua\@undefined
  \ifx\widowpenalties\@undefined
    \mathchardef\e@alloc@top=255
    \let\e@alloc@chardef\chardef
  \else
    \mathchardef\e@alloc@top=32767
    \let\e@alloc@chardef\mathchardef
  \fi
\else
  \chardef\e@alloc@top=65535
  \let\e@alloc@chardef\chardef
\fi
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\e@alloc@chardef}{Extended Allocation}%
%<latexrelease>\let\e@alloc@top\@undefined
%<latexrelease>\let\e@alloc@chardef\@undefined
%<latexrelease>\EndIncludeInRelease
```

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

```
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\e@mathgroup@top}{Extended Allocation}%
\ifx\Umathcode\@undefined
  \chardef\e@mathgroup@top=16
\else
  \chardef\e@mathgroup@top=256
\fi
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\e@mathgroup@top}{Extended Allocation}%
%<latexrelease>\let\e@mathgroup@top\@undefined
%<latexrelease>\EndIncludeInRelease
```

`\e@alloc` A modified version of `\alloc@` that takes the count register rather than just the final digit of its number (assuming `\count`1x). It also has an extra argument to give the top of the extended range.

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

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

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

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

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

(*End definition for* `\e@alloc`.)

`\e@ch@ck` Extended check command. If the first range is exceeded, bump to 256 (or 266 for counts) Allocate matching registers from the top of the extended range and add to `\@freelist`.

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/10/01}
%<latexrelease>                 {\e@ch@ck}{Extended Allocation (checking)}%
```

```
\gdef\e@ch@ck#1#2#3#4{%
  \ifnum#1<#2\else
```

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

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

Check we are below the extended limit. Add `\string` in case argument is not an unexpandable primitive

```
    \ifnum#1<#3\relax
    \else
      \errmessage{No room for a new \string#4}%
    \fi
  \fi}%
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\e@ch@ck}{Extended Allocation (checking)}%
%<latexrelease>\gdef\e@ch@ck#1#2#3#4{%
%<latexrelease>  \ifnum#1<#2\else
%<latexrelease>    \ifnum#1=#2\relax
%<latexrelease>      #1\@cclvi
%<latexrelease>      \ifx\count#4\advance#1 10 \fi
%<latexrelease>    \fi
%<latexrelease>    \ifnum#1<#3\relax
```

```
%<latexrelease>    \else
%<latexrelease>      \errmessage{No room for a new #4}%
%<latexrelease>    \fi
%<latexrelease>  \fi}%
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\e@ch@ck}{Extended Allocation (checking)}%
%<latexrelease>\let\e@ch@ck\@undefined
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\extrafloats}{Extra floats}%
```

(*End definition for* `\e@ch@ck`.)

```
\let\float@count\e@alloc@top
\ifx\numexpr\@undefined
```

`\extrafloats`

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

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

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

\alloc@

```
\def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
  \ch@ck#1#4#2%
  \allocationnumber\count1#1%
  \global#3#5\allocationnumber
  \wlog{\string#5=\string#2\the\allocationnumber}}
```

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

\newinsert{⟨*cmd*⟩}

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/10/01}
%<latexrelease>                 {\newinsert}{Extended ↩
      ↪\newinsert}%
```

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

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

```
\else
```

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

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

```
\def\newinsert#1{%
\@tempswafalse
\global\advance\insc@unt\m@ne
\ifnum\count10<\insc@unt
\ifnum\count11<\insc@unt
\ifnum\count12<\insc@unt
\ifnum\count14<\insc@unt
  \@tempswatrue
```

```
\fi\fi\fi\fi
\if@tempswa
\allocationnumber\insc@unt
\else
\global\advance\insc@unt\@ne
  \extrafloats\@ne
  \@next\@currbox\@freelist
    {\ifnum\@currbox<\e@insert@top
      \allocationnumber\@currbox
     \else
     \ch@ck0\m@ne\insert
     \fi}%
     {\ch@ck0\m@ne\insert}%
\fi
\global\chardef#1\allocationnumber
\wlog{\string#1=\string\insert\the\allocationnumber}%
}
```

```
\fi
%</2ekernel|latexrelease>
```

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

`\ch@ck{}`

`{}{type}` The dreadful error message, especially if `\#3` is `write`. Thanks to LuaTeX not seen often.

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

`\newhelp`

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

(*End definition for* `\newhelp`.)

`\maxdimen`
`\hideskip`

Here are some examples of allocation.

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

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

`\p@`

`\z@`
`\z@skip`
`\voidb@x`

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

(*End definition for* `\z@`, `\z@skip`, *and* `\voidb@x`.)

Assign initial values to TEX's parameters

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

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

```
\pretolerance=100
\tolerance=200 % INITEX sets this to 10000
\hbadness=1000
\vbadness=1000
\linepenalty=10
\hyphenpenalty=50
\exhyphenpenalty=50
\binoppenalty=700
\relpenalty=500
\clubpenalty=150
\widowpenalty=150
\displaywidowpenalty=50
\brokenpenalty=100
\predisplaypenalty=10000
\doublehyphendemerits=10000
\finalhyphendemerits=5000
\adjdemerits=10000
\tracinglostchars=1
\uchyph=1
\defaulthyphenchar=`\-
\defaultskewchar=-1
\delimiterfactor=901
```

In LATEX we don't want box information in the transcript unless we do a full tracing.

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

```
\hfuzz=0.1pt
\vfuzz=0.1pt
\overfullrule=5pt
\maxdepth=4pt
\splitmaxdepth=\maxdimen
\boxmaxdepth=\maxdimen
\delimitershortfall=5pt
\nulldelimiterspace=1.2pt
\scriptspace=0.5pt
\parindent=20pt
\parskip=0pt plus 1pt
\abovedisplayskip=12pt plus 3pt minus 9pt
\abovedisplayshortskip=0pt plus 3pt
\belowdisplayskip=12pt plus 3pt minus 9pt
\belowdisplayshortskip=7pt plus 3pt minus 4pt
\topskip=10pt
\splittopskip=10pt
\parfillskip=0pt plus 1fil
```

\normalbaselineskip
\normallineskip
\normallineskiplimit

We also define special registers that function like parameters:

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

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

\interfootlinepenalty

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

Definitions for preloaded fonts

\magstephalf
\magstep

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

(*End definition for* `\magstephalf` *and* `\magstep`.)

2.2 Macros for setting ordinary text

\frenchspacing

\nonfrenchspacing

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

\normalbaselines

Set all baselines to their "normal" values.

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

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

\lq
\rq

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

(*End definition for* \lq *and* \rq.)

\lbrack

\rbrack

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

\aa
\AA

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

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

(*End definition for* \aa *and* \AA.)

\endgraf

As usual a control sequence is let equal to **\par**

```
\let\endgraf=\par
```

\endline

```
\let\endline=\cr
```

(*End definition for* \endline.)

\space

A very simple definition of a space

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

\empty

The primitive **\empty** is let to the LaTeX equivalent.

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

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

\null

The **\null** macro is just an empty **\hbox**.

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

\bgroup
\egroup

Provide macros for the left and right braces, denoting grouping.

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

(*End definition for* `\bgroup` *and* `\egroup`.)

\obeylines

The macros here are identical to those in plain TeX. In **\obeylines**, we say **\let^^M=\par** instead of **\def^^M{\par}** since this allows, for example, **\let\par=\cr \obeylines \halign{...** in which **\cr**'s need not be given in the alignment.

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

\obeyspaces

A switch to typeset spaces verbatim. Used in verbatim environments.

```
\def\obeyspaces{\catcode`\ \active}
{\obeyspaces\global\let =\space}
```

\loop
\iterate
\repeat

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

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

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

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

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

LaTeX defines **\smallskip**, etc. in **ltspace.dtx** (See **??**).

\nointerlineskip
\offinterlineskip

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

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

The next two macros can be used to introduce "spacing" by means of a **\vrule** or **\hrule**. These are useful for introducing spacing at the start of a line or the top and bottom part of the page, where normal skips will fail. LaTeX discourages these and defines **\hspace** and **\vspace**.[1]

\hglue{⟨*glue specification*⟩}

```
\def\hglue{\afterassignment\hgl@\skip@=}
\def\hgl@{\leavevmode \count@\spacefactor \vrule \@width\z@
  \nobreak\hskip\skip@ \spacefactor\count@}
```

\vglue{⟨*glue specification*⟩}

```
\def\vglue{\afterassignment\vgl@\skip@=}
\def\vgl@{\par \dimen@\prevdepth \hrule \@height\z@
  \nobreak\vskip\skip@ \prevdepth\dimen@}
```

Example 2.1 hglue

```
A\hglue 10pt plus1pt minus0.5pt test example.

\hglue 50pt ;test
```

A test example.
;test

LaTeX defines ~ in **ltdefns.dtx**, see page **??**.

\slash

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

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

The following macros introduce penalty markers that make break less, or more, desirable. The macros **\break**, **\nobreak** and **\allowbreak** set penalties to particular values. In the original plain TeX, penalties were numbers. In LaTeX 2e were changed to macros for efficiency.

\break

Forces a break

```
\def\break{\penalty-\@M}
```

\nobreak

[1] See https://tex.stackexchange.com/questions/30048/what-is-the-difference-between-vskip-and-vspace

\allowbreak

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

(*End definition for* `\allowbreak`.)

\filbreak
\goodbreak

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

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

\eject

Define **\eject** as in plain TEX but define **\supereject** only in the compatibility file.[2]

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

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

\removelastskip

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

(*End definition for* `\removelastskip`.)

Example 2.2 Removing the lastskip

```
CCCC\par
\medskip
\the\lastskip
\medskip
\removelastskip
\the\lastskip
DDDD
```

CCCC

0.0pt0.0ptDDDD

Adding Penalties The

\smallbreak
\medbreak
\bigbreak

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

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

[2] v1.1s 1995/10/17 Move **\supereject** to compat file

\m@th

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

(*End definition for* `\m@th`.)

\underbar

Due to LaTeX's redefinition of **\underline** plain TeX's **\underbar** can be done in a simpler fashion (but do we need it at all?). [3] [4]

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

(*End definition for* `\underbar`.)

\strutbox

In plain TeX **\strut** is implemented as a rule of width zero, since this takes minimum space and time in applications where numerous struts are present

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

\strut

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

The **\relax** TeXBook [353] is necessary in case **\strut** appears first in an alignment entry, because TeX is in somewhat unpredictable mode at such times (see Chapter 22 in the TeXBook.

(*End definition for* `\strutbox` *and* `\strut`.)

\hidewidth

For alignment entries that can stick out.

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

(*End definition for* `\hidewidth`.)

At this point in plain TeX we find the definitions of **\item** and **\itemitem**. These were removed in LaTeX 2e obviously since we have lists. [5] [6]Move **\hang** and **\textindent** to latex209.def [7]

\narrower

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

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

Most text commands are actually encoding specific and therefore defined later, so commented out or removed from this file.

\leavevmode

begins a paragraph, if necessary

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

[3] v1.1m 1994/05/26 (CAR/FMi) changed to use box **\tw@**
[4] v1.1p 1994/05/26 (DPC) changed to use **\sbox**
[5] v1.0h 1994/05/16 Remove unnecessary def for **\item**
[6] 1i 1995/04/27
[7] RmS 1991/11/04 Removed **\itemitem** since neverneeded/useful with LaTeX.

\mathhexbox

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

(*End definition for* `\mathhexbox`.)

\ialign

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

(*End definition for* `\ialign`.)

\oalign

\o@lign

\ooalign

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

\sh@ft

The definition of this macro in plain.tex was improved in about 1997; but as a result its usage was changed and its new definition is not appropriate for LaTeX.

Since the version given here has been in use by LaTeX for many years it does not seem prudent to remove it now. As far as we can tell it has only been used to define **\b** and **\d** but this cannot be certain.

```
\def\sh@ft#1{\dimen@.00#1ex\multiply\dimen@\fontdimen1\font
  \kern-.0156\dimen@} % compensate for slant in lowered accents
\def\ltx@sh@ft #1{%
  \dimen@ #1%
  \kern \strip@pt
    \fontdimen1\font \dimen@
  } % kern by #1 times the current slant
```

\hrulefill

\dotfill

LaTeX change: **\kern\z@** added to end of **\hrulefill** and **\dotfill** to make them work in 'tabular' and 'array' environments. [8] LaTeX change: **\leavevmode** added at beginning of **\dotfill** and **\hrulefill** so that they work as expected in vertical mode.

[8] Change made 24 July 1987.

```
\def\hrulefill{\leavevmode\leaders\hrule\hfill\kern\z@}
```

The box in **\dotfill** originally contained (in plain.tex):
\mkern 1.5mu .\mkern 1.5mu;
the width of .44em differs from this by .04pt which is probably an acceptable difference within leaders.

```
\def\dotfill{%
  \leavevmode
  \cleaders \hb@xt@ .44em{\hss.\hss}\hfill
  \kern\z@}
\sfcode`\)=0 \sfcode`\'=0 \sfcode`\]=0
```

2.3 Definitions related to output

\magnification doesn't work in LaTeXthis has been set to verbatim in the code.

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

\showoverfull The following commands are used in debugging:

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

(*End definition for* `\showoverfull`.)

\showoutput

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

(*End definition for* `\showoutput`.)

```
%<latexrelease>\IncludeInRelease{2015/01/20}{\loggingall}{etex ⤸
      (tracing}%
%<*2ekernel|latexrelease>
\ifx\tracingscantokens\@undefined
\gdef\loggingall{%
  \tracingstats\tw@
  \tracingpages\@ne
  \tracinglostchars\@ne
  \tracingparagraphs\@ne
  \errorcontextlines\maxdimen
  \loggingoutput
  \tracingmacros\tw@
  \tracingcommands\tw@
  \tracingrestores\@ne
  }%
\else
\gdef\loggingall{%
  \tracingstats\tw@
  \tracingpages\@ne
```

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

\tracingnone
\hideoutput

```
%<latexrelease>\IncludeInRelease{2015/01/20}{\tracingnone}%
%<latexrelease>                              {turn off etex ↊
      ↋tracing}%
%<*2ekernel|latexrelease>
\ifx\tracingscantokens\@undefined
\def\tracingnone{%
  \tracingonline\z@
  \tracingcommands\z@
  \showboxdepth\m@ne
  \showboxbreadth\m@ne
  \tracingoutput\z@
  \errorcontextlines\m@ne
  \tracingrestores\z@
  \tracingparagraphs\z@
  \tracingmacros\z@
  \tracinglostchars\@ne
  \tracingpages\z@
  \tracingstats\z@
}%
\else
\def\tracingnone{%
  \tracingassigns\z@
  \tracingrestores\z@
  \tracingonline\z@
  \tracingcommands\z@
  \showboxdepth\m@ne
```

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

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

```
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\tracingnone}%
%<latexrelease>                              {turn off etex ↙
      ↖tracing}%
%<latexrelease>\let\tracingnone\@undefined
%<latexrelease>\let\hideoutput\@undefined
%<latexrelease>\EndIncludeInRelease
```

(*End definition for* `\tracingnone` *and* `\hideoutput`.)

LaTeX change: **\showhyphens** Defined later. Punctuation affects the spacing. The **\nonfrenchspacing** is set. This can be overwritten later on by the user, if required.

```
%<*2ekernel>
\nonfrenchspacing
%</2ekernel>
```

h a n g

3

Version Identification

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

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

\fmtname
\fmtversion
\latexreleaseversion
\patch@level

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

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

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

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

```
%%<*2ekernel>
\def\fmtname{LaTeX2e}
\edef\fmtversion
%%</2ekernel>
%%<latexrelease>\edef\latexreleaseversion
%%<*2ekernel|latexrelease>
   {2018-12-01}
%%</2ekernel|latexrelease>
%%<*2ekernel>
\def\patch@level{0}
```

(*End 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:-)).

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

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

```
\ifnum\count@>65
  \typeout{^^J%
!!!
```

```
! If you enter <return> to scroll past this message then the format^^J%
! will be built, but please consider obtaining newer source files^^J%
! before continuing to build LaTeX.^^J%
!!!
```

```
\def\@IncludeInRele@se#1#2#3{%
  \toks@{[#1] #3}%
  \expandafter\ifx\csname\string#2+\@currname+IIR\endcsname\relax
    \ifnum\expandafter\@parse@version#1//00\@nil
          >\expandafter\@parse@version\fmtversion//00\@nil
      \GenericInfo{}{Skipping: \the\toks@}%
     \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
    \else
      \GenericInfo{}{Applying: \the\toks@}%
      \@includeinreleasetrue
      \expandafter\let\csname\string#2+\@currname+IIR\endcsname\@empty
    \fi
  \else
    \GenericInfo{}{Already applied: \the\toks@}%
    \expandafter\@gobble@IncludeInRelease
  \fi
}
\def\EndIncludeInRelease{%
\if@includeinrelease
  \@includeinreleasefalse
\else
  \PackageError{latexrelease}{mis-matched ↩
        ↪EndIncludeInRelease}{}%
\fi}
\long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{%
  \@includeinreleasefalse
  \@check@IncludeInRelease#1\IncludeInRelease\@check@IncludeInRelease
  \@end@check@IncludeInRelease}
\long\def\@check@IncludeInRelease#1\IncludeInRelease
                                  #2#3\@end@check@IncludeInRelease{%
  \ifx\@check@IncludeInRelease#2\else
    \PackageError{latexrelease}{skipped IncludeInRelease}{}%
  \fi}
%</2ekernel|latexrelease>
```

(*End definition for* `\IncludeInRelease`.)

h a n g

4

LaTeXe Kernel Definitions Module

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

4.1 Introduction

This section contains a number of commands used in defining other macros, as well as some useseful commands, that can be used by package authors. Some of these commands such as `\@height` were defined to save tokens and hence memory tokens and by now they do not add much value to a modern LaTeX installation. However, removing them would break backward compatibility, but perhaps in your own packages you may choose to improve the program readability by not using them.

?? → P. ??	prefix a number less than 10 with '0'
`\typeout`	display something on the terminal
`\newlinechar`	newline character
`\@height`	height
`\@width`	width
`\@depth`	depth

`\two@digits{number}`

Prefix a number less than 10 with '0'.

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

`\typeout`⟨*message*⟩

Display something on the terminal.

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

`\newlinechar`

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

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

4.2 Saved versions of TeX primitives

`\@@par`

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

```
\let\@@par=\par
\let\@@input=\input     %%% moved earlier
\let\@@end=\end         %%%
```

`\@@hyph`

\-

The following comment was added when these commands were first set up, 19 April 1986: the `\-` command is redefined to allow it to work in the `\ttfamily` type style, where automatic hyphenation is suppressed by setting `\hyphenchar` to −1. The original primitive TEX definition is saved as `\@@hyph` just in case anyone needs it.
There is a need for a robust command for a discretionary hyphen since its exact representation depends on the glyphs available in the current font. For example, with suitable fonts and the `T1` font encoding it is possible to use hanging hyphens.
A suitable robust definition that allows for many possible types of font and encoding may be as follows:

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

The redefinition (via `\let`) of `\-` within tabbing also makes the use of a robust command advisable since then any redefinition of `\-` via `\DeclareRobustCommand` will not cause a conflict.
Therefore, macro writers should be hereby warned that these internals will probably change! It is likely that a future release of LATEX will make `\-` effectively an encoding specific text command.

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

\@dischyph

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

\@@italiccorr

Save the original italic correction.

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

\@height

\@depth

\@width

\@minus

\@plus

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

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

\hb@xt@

The next one is another 100 tokens worth.

```
\def\hb@xt@{\hbox to}
\message{hacks,}
```

4.2.1 Command definitions

This section defines the following commands:

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

\@nameuse {⟨*name*⟩} Expands to ****⟨*name*⟩.

\@ifnextchar X{⟨*Yes*⟩}{⟨*No*⟩} Expands to ⟨*Yes*⟩ if next character is an 'X', and to ⟨*No*⟩ otherwise. (Uses **\reserved@a**–**\reserved@c**.) NoTE: GOBBLES ANY SPACE FOLLOWING IT.

\@ifstar {⟨*Yes*⟩}{⟨*No*⟩} Gobbles following spaces and then tests if the next character is a '*'. If it is, then it gobbles the star ('*') and expands to ⟨*Yes*⟩, otherwise it expands to ⟨*No*⟩.

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

\@ifundefined {⟨*name*⟩}{⟨*Yes*⟩}{⟨*No*⟩}
: If **\name** is undefined then it executes ⟨*Yes*⟩, otherwise it executes ⟨*No*⟩. More precisely, true if **\name** either undefined or = **\relax**.

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

\newcommand *{⟨**\FOO**⟩}[⟨*i*⟩]{⟨*TEXT*⟩}
User command to define **\FOO** to be a macro with i arguments (i = 0 if missing) having the definition ⟨*TEXT*⟩. Produces an error if **\FOO** already defined.

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

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

`\newenvironment` `*{⟨FOO⟩}[⟨i⟩]{⟨DEF1⟩}{⟨DEF2⟩}`
equivalent to:
`\newcommand{\FOO}[i]{DEF1} \def{\endFOO}{DEF2}`
(or the appropriate star forms).

`\renewenvironment` Obvious companion to `\newenvironment`.

4.2.2 \@cons and \@car

The following macros are used extensively for lists.

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

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

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

`\typeout` `{⟨message⟩}`
Produces a warning message on the terminal.

`\typein` `{⟨message⟩}`
Types message, asks the user to type in a command, then executes it.

`\typein` `[⟨\CS⟩]{⟨MSG⟩}`
Same as above, except defines `\CS` to be the input instead of executing it.[1]

4.3 Code

typein

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

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

(*End definition for* `typein`.)

[1] LaTeX209 1992/03/18 (RMS) changed input channel from 0 to `\@inputcheck` to avoid conflicts with other channels allocated by `\newread`

4.3.1 csname abstractions

\@namedef

The **??**[→P.??] and **??**[→P.??] are used to define or use **\csname**. You need to watch that the arguments are *names* i.e, they are not preceded by a backslash or other active character. (See example **??**).

\@nameuse

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

Example 3.1 Using \@namedef

```
\makeatletter
\@namedef{test1}{This is the first test}
\@nameuse{test1}
\makeatother
```

This is the first test

4.4 List helper macros

These helper macros are primarily for constructing and manipulating lists.

\@cons

The command **\@cons** is used for the construction of lists. Its unique characteristic being that it constsructs *elt* lists. What this means is that each entry in the list is preceded by another as yet undefined command **\@elt**. This can be later used to grub the element as its argument and do something about it.

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

Assume we have a list **\mylist** which has been defined as **\mylist{\@elt one \@elt two ...}**. We can use **\@cons** to append more elements as shown in the next example,

Example 4.2 Usage of cons

```
\makeatletter

% define a list
\def\mylist{}

% add elements
\@cons\mylist{{one}}
```

```
\@cons\mylist{{two}}

% define \@elt to be a space
\def\@elt{\space}

% type the meaning
\meaning\mylist

% define \@elt to print its argument with a comma
\def\@elt#1{#1, }

% print the list
\mylist

\makeatother
```

macro:->\@elt {one}\@elt {two}
one, two,

If you have noticed, when we added the element, we have used an extra set of brackets. If we did not do this `\@elt` would have catpured only the first letter taht follows it and the result would have been `o, net, wo`. Try it out.

`\@car`

`\@cdr`

The next two macros `\@car` and `\@cdr` capture either the first element of a list or the rest of the elements except the first. They are both delimited by `\@nil`.

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

`\@carcube` Given a list $T_1 \ldots T_n$ `\@carcube` will grub $T_1 T_2 T_3$ for `T1 T2 T3` , $n > 3$. If you are familiar with lisp `car`, `cdr` and `cons` are fundamental functions. The `cons` function is used to construct lists, and the `car` and `cdr` functions are used to take them apart.[2]

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

(*End definition for* `\@carcube`.)

[2]The name of the cons function is not unreasonable: it is an abbreviation of the word 'construct'. The origins of the names for `car` and `cdr`, on the other hand, are esoteric: `car` is an acronym from the phrase 'Contents of the Address part of the Register'; and cdr (pronounced 'could-er') is an acronym from the phrase 'Contents of the Decrement part of the Register'. These phrases refer to specific pieces of hardware on the very early computer on which the original Lisp was developed. Besides being obsolete, the phrases have been completely irrelevant for more than 35 years to anyone thinking about Lisp. Nonetheless, although a few brave scholars have begun to use more reasonable names for these functions, the old terms are still in use. In particular, since the terms are still used in the Emacs Lisp source code.

\@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

\@preamblecmds

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

\@onlypreamble\@onlypreamble
\@onlypreamble\@preamblecmds
```

4.5 Command Building

\@star@or@long ⟨*parameter text*⟩

This is an interesting command in that it checks if #1 is a star then sets **??** → P. **??** to either **\long** or **\relax**. It is used for building **??** → P. **??**, which is always **\long** except its star version.

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

\l@ngrel@x

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

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

\newcommand {⟨*cmd name*⟩} [⟨*number of arguments*⟩] [⟨*default for optional argument*⟩] {⟨*replacement text*⟩}

User level **\newcommand**.

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

The command simply checks if it is long or short and the calls **??** → P. **??**.

\new@command

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

\@newcommand

\@argdef

\@xargdef

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

Define **#1** if it is definable.
Both here and in **\@xargdef** the replacement text is absorbed as an argument because if we are not allowed to make the definition we have to get rid of it completely.

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

```
\long\def\@xargdef#1[#2][#3]#4{%
  \@ifdefinable#1{%
```

Define the actual command to be:
\def\foo{\@protected@testopt\foo\\foo{default}}
where **\\foo** is a csname generated from applying **\csname** and **\string** to **\foo**, ie the actual name contains a backslash and therefore can't clash easily with existing command names. "Default" is the contents of the second optional argument of **(re)newcommand**.

Example 5.3 newcommand internal form

```
        \newcommand\mytest[2][other material] {some
            text}
        \meaning\mytest
```

macro:->\@protected@testopt \mytest \\mytest {other material}

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

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

```
        \expandafter\@yargdef
          \csname\string#1\endcsname
           \tw@
           {#2}%
           {#4}}}
```

\@testopt

This macro encapsulates the most common call to **\@ifnextchar**, saving several tokens each time it is used in the definition of a command with an optional argument.

#1 The code to execute in the case that there is a [need not be a single token but can be any sequence of commands that 'expects' to be followed by [. If this command were only used in **\newcommand** definitions then **#1** would be a single token and the braces could be omitted from **{#1}** in the definition below, saving a bit of memory.

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

\@protected@testopt

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

This method of making commands robust avoids the need for using up two csnames per command, the price is the extra expansion time for the **\ifx** test.

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

\@yargdef

\@yargd@f

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

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

Recall that LaTeX2.09 goes into an infinite loop with **\renewcommand[1]{\@tempa}{foo}** (DPC 6 October 93).

Reorganised again (DPC 1999). Rather than make a loop to construct the argument spec by counting, just extract the required argument spec by using a delimited argument (delimited by the digit). This is faster and uses less tokens. The coding is slightly odd to preserve the old interface (using **#2** = **\tw@** as the flag to surround the first argument with **[]**. But the new method did not allow for the number of arguments **#3** not being given as an explicit digit; hence (further expansion of this argument and use of) **\number** was added later in 1999.

It is not clear why these are still **\long**.

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

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

\@reargdef

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

\renewcommand

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

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

\renew@command

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

4.5.1 Checking if a command is definable

\@ifdefinable

Test if user is allowed to define a command.

\@rc@ifdefinable

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

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

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

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

This command is not a general command for package builders but it has its uses. If the command has been defined then it issues an error message via `\@notdefinable`. In the example below we save the `\@notdefinable` in order to redirect the error message to the example. We check if the counter `c@chapter` can be defined, since it has it will activate the `\@notdefinable` and print an error. We then restore the command to its previous meaning.

Example 5.4 ifdefinable

```
\makeatletter
\let\save@notdefinable\@notdefinable
\def\@notdefinable{Not definable}
\@ifdefinable{c@chapter}{true}
\let\@notdefinable\save@notdefinable
\makeatother
```

Not definable

4.6 Environment Building commands

`\newenvironment`

Define a new user environment. `#1` is the environment name. `#2#` Grabs all the tokens up to the first `{`. These will be any optional arguments. They are not parsed at this point, but are just passed to `\@newenv` which will eventually call `\newcommand`. Any optional arguments will then be parsed by `\newcommand` as it defines the command that executes the 'begin code' of the environment.
This `#2#` trick removed with version 1.2i as it fails if a `{` occurs in the optional argument. Now use `\@ifnextchar` directly.

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

`\new@environment`

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

`\@newenva`

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

\@newenvb

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

\renewenvironment

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

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

\renew@environment

As a **csname** is used the name of an envronment can contain non-letters, such as dashes etc.

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

\@newenv

The internal version of **\newenvironment**.

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

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

```
\long\def\@newenv#1#2#3#4{%
  \@ifundefined{#1}%
    {\expandafter\let\csname#1\expandafter\endcsname
                          \csname end#1\endcsname}%
    \relax
  \expandafter\new@command
     \csname #1\endcsname#2{#3}%
%<autoload>\aut@global
     \l@ngrel@x\expandafter\def\csname end#1\endcsname{#4}}
```

4.7 newif

The LaTeX kernel provides the **??**[→ P. ??] command that can be used to define boolean switches. It uses a couple of tricks to remove the **if** part of the command being defined and to **\let** the switches to true or false.

\newif{⟨*cmd*⟩}

This is as the notes in the kernel mention a different type of allocation. A macro to define boolean switches

For example, **\newif\if@foo** creates **\@footrue**, **\@foofalse** to go with **\if@foo**.[3]

Before we describe how this is achieved we make a small digression to test a not so widely known trick to get the **\string** not to print the backslash character, by setting the **\escapechar** to -1. This is really an esoteric trick.

Example 7.5 escapechar

```
\bgroup
\def\test{This is a test}
\makeatletter
\escapechar\m@ne
\test

\string\test

test
\makeatother
\egroup
```

This is a test
test
test

The **\newif** is defined in a very clever way.

```
\def\newif#1{%
  % save original definition
  \count@\escapechar

  % allocate to -1
  \escapechar\m@ne

  % set to false as a default
    \let#1\iffalse

  % define \@footrue and \@foofalse
    \@if#1\iftrue
    \@if#1\iffalse
  \escapechar\count@}
```

\@if⟨*name of newif*⟩⟨**ifalse or iftrue**⟩

The auxiliary macro for the **\newif** command.

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

This is a mind twisting way to define the **\newif** and I am sure a method applauded by every Byzantine General. It also resolved a long standing query

[3] Customarily these command use the '@' to make the definitions internal and also more readable.

in my head as to why there was never a `@gobblethree` command in the kernel (see **??**).

4.8 Provide version of commands

\providecommand

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

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

\provide@command

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

\CheckCommand

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

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

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

```
\@onlypreamble\CheckCommand
```

\check@command

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

\@check@c

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

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

`\@check@eq`

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

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

4.8.1 Argument gobbling command utilities

The `\@gobble` macro is used to get rid of its argument. Similarly `\@gobbletwo` and `\@gobblefour` eat two or four arguments respectively.

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

4.8.2 Other argument grabbers

The commands `\@firstofone`, `\@firstoftwo` and `\@secondoftwo` are amongst the most widely used commands in the kernel. They only grab the argumeny indicated by their name.

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

`\@iden`

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

`\@iden`

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

(*End definition for* `\@iden`.)

`\@thirdofthree`

Another grabber now used in the encoding specific section.

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

\@expandtwoargs

A macro to totally expand two arguments to another macro

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

Example 8.6 Expand two arguments

```
\makeatletter
\bgroup
  \def\test#1#2{#1, #2}
  \def\xx{first argument}
  \def\yy{second argument}
  \@expandtwoargs\test{\xx}{\yy}
\egroup
\makeatother
```

first argument, second argument

\@backslashchar

A category code 12 backslash. See also **??**→ P. ?? which is a better alternative.

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

4.9 Robust commands and protect

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

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

The method LaTeX uses for making fragile commands robust is to precede them with `\protect`. This can have one of five possible values:

- `\relax`, for normal typesetting. So `\protect\foo` will execute `\foo`.
- `\string`, for writing to the screen. So `\protect\foo` will write `\foo`.
- `\noexpand`, for writing to a file. So `\protect\foo` will write `\foo` followed by a space.

- `\@unexpandable@protect`, for writing a moving argument to a file. So `\protect\foo` will write `\protect\foo` followed by a space. This value is also used inside `\edef`s, `\mark`s and other commands which evaluate their arguments fully.
- `\@unexpandable@noexpand`, for performing a deferred write inside an `\edef`. So `\protect\foo` will write `\foo` followed by a space. If you want `\protect\foo` to be written, you should use `\@unexpandable@protect`. (Removed as never used).

`\@unexpandable@protect`

`\@unexpandable@noexpand`

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

`\DeclareRobustCommand`

`\declare@robustcommand`

This is a package-writers command, which has the same syntax as `\newcommand`, but which declares a protected command. It does this by having
`\DeclareRobustCommand\foo`
define `\foo` to be `\protect\foo<space>`,
and then use `\newcommand\foo<space>`.
Since the internal command is `\foo<space>`, when it is written to an auxiliary file, it will appear as `\foo`.
We have to be a bit cleverer if we're defining a short command, such as `\_`, in order to make sure that the auxiliary file does not include a space after the command, since `\_ a` and `\_a` aren't the same. In this case we define `\_` to be:

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

which expands to:

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

Then if `\protect` is `\@typeset@protect` (normally `\relax`) then we just perform `\_<space>`, and otherwise `\@x@protect@` gobbles everything up and expands to `\protect\_`.
Note: setting `\protect` to any value other than `\relax` whilst in 'typesetting' mode will cause commands to go into an infinite loop! In particular, setting `\relax` to `\@empty` will cause `\_` to loop forever. It will also break lots of other things, such as protected `\ifmmode`s inside `\halign`s. If you really really have to do such a thing, then please set `\@typeset@protect` to be `\@empty` as well. (This is what the code for `\patterns` does, for example.)
More fun with `\expandafter` and `\csname`.

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

`\@x@protect`

`\x@protect`

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

`\@typeset@protect`

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

`\set@display@protect`

`\set@typeset@protect`

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

`\protected@edef`

`\protected@xdef`

`\unrestored@protected@xdef`

\restore@protect

The commands \protected@edef and \protected@xdef perform 'safe' \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!

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

\protect

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

4.9.1 Internal defining commands

\@ifundefined

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

\@qend

\@qrelax

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

\@ifnextchar

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

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

\kernel@ifnextchar

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

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

\@ifnch

\@ifnch is a tricky macro to skip any space tokens that may appear before the character in question. If it encounters a space token, it calls **\@xifnch**.[4]

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

\@sptoken

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

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

[4] There is an error in the kernel at this point in the documentation, as the verbatim guards have been omitted.

Example 9.7 sptoken

```
\makeatletter
\noindent
A\:\:A\@sptoken\@sptoken A\\
AAA
\makeatother
```

A A A
AAA

\@xifnch

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

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

The following two commands are widely used in preambles to change the **\catcode** of **@** to 11 or 12.

\makeatletter

Make internal control sequences accessible and inaccessible (**\makeatother**).

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

This **\@ifstar** command is another useful

\@ifstar{⟨*true code*⟩} {⟨*false code*⟩}

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

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

\@dblarg

\@xdblarg

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

\@sanitize

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

```
\def\@sanitize{\@makeother\ ⤸
      ⤹\@makeother\\\@makeother\$\@makeother\&%
\@makeother\#\@makeother\^\@makeother\_\@makeother\%\@makeother\~}
```

\@onelevel@sanitize

This makes the whole "meaning" of **#1** (its one-level expansion) into catcode 12 tokens: it could be used in **\DeclareRobustCommand**.
If it is to be used on default float specifiers, this should be done when they are defined.[5]

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

4.9.2 Commands for Autoloading

\aut@global

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

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

\@autoload

Use **\@@input** not **\input** to save string space and stops autoload files appearing in **\listfiles**
This macro is only defined in the 'autoload' format. It inputs a package '**auto#1.sty**' within a local group, and with normalised catcodes. **\aut@global** is set to **\global** so that **\newif** **\newcommand** and related commands make global definitions.

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

This been a rather long discussion, but this is almost the heart of the kernel commands that are most useful for package writers. Spend time to understand this section well, as many of the hacks described here are useful and are used in most of the other sections of the kernel.

[5] v1.2c 1994/10/30 Macro added
[6] v1.2q 1995/10/02 Macro added

h a n g

5

Allocations

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

5.1 Counters

This section deals with counter and other variable allocation.

```
%<*2ekernel>
```

The following are from plain TEX:

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

\@ne The number 1.

\m@ne The number −1.

\tw@ The number 2.

\sixt@@n The number 16.

\@m The number 1000.

\@MM The number 20000.

\@xxxii The constant 32.

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

(*End definition for* `\@xxxii`.)

\@Mi **\@Mii** **\@Miii** **\@miv** Constants 1001–1004. These are used mostly for special penalty values used in the output routines to signal the context of the caller.

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

(*End definition for* `\@Mi` *and others.*)

5.2 Scratch registers

\@tempcnta **\@tempcntb** Scratch count registers used by LATEX kernel commands.

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

(*End definition for* `\@tempcnta` *and* `\@tempcntb`.)

\if@tempswa General boolean switch used by LATEX kernel commands. To remmeber the name think of temporary switch 'a'. This is used extensively when writing to auxiliary, toc and other list files.

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

(*End definition for* `\if@tempswa`.)

\@tempdima **\@tempdimb** **\@tempdimc** Scratch dimen registers used by LATEX kernel commands.

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

(*End definition for* `\@tempdima`, `\@tempdimb`, *and* `\@tempdimc`.)

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

```
\newbox\@tempboxa
```

(*End definition for* `\@tempboxa`.)

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

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

(*End definition for* `\@tempskipa` *and* `\@tempskipb`.)

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

```
\newtoks\@temptokena
```

(*End definition for* `\@temptokena`.)

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

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

(*End definition for* `\@flushglue`.)

```
%</2ekernel>
```

h a n g

6

Control Structures

In antiquity men and women saw each other as different; accordingly, they developed complex taxonomies (philosophical explanations) for understanding anatomical, physiological, emotional, and rational differences.
Some of these differences seem profoundly odd to us moderns.
Modern discussions about erotic art have often concerned the place of women: to what extent are they objects of social manipulation, to what extent can they be subjects?

21.00002
16
-2.5
-2.1
0.87865

6.1 Introduction

Looping structures are an absolute necessity to handle lists and other constructions. As TEX is so different from traditional programming languages, macro writers propose to harnass TEX into a more familiar system, by imposing syntaxes borrowed from various successful highlevel programming languages.

laan1992 claims that this sugaring does injustice to TEX's nature might result, and users might become intimidated, because of the difficult—at least unusual— encoding used to achieve the aim. The more so when functional equivalents are already there, although perhaps hidden, and not tagged by familiar names.[1]

Leslie Lamport provided a somewhat limited amount of macros, but adequate for LATEX. It must be emphasized that **expl3** did a tremendous job in providing tens of alternatives in a more organized manner. If you need to program any serious amount of code you are advised to move over to these. The **etoolbox** package[2] also provides many alternatives. However it is advisable to go over these older methods in order to understand the rest of the kernel.[3]

```
%<*2ekernel>
\message{control,}
```

`\@whilenoop`

`\@whilenum`

`\@iwhilenum`

Changes [4] [5] [6]

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

`\@whiledim`

```
\long\def\@whiledim#1\do #2{\ifdim ⤸
      ⤹#1\relax#2\@iwhiledim{#1\relax#2}\fi}
```

`\@iwhiledim`

```
\long\def\@iwhiledim#1{\ifdim #1\expandafter\@iwhiledim
        \else\expandafter\@gobble\fi{#1}}
```

`\@whileswnoop`

`\@whilesw`

[1] **laan1992**.
[2] **etoolbox**.
[3] **Kabelschacht1987**.
[4] v1.0f 1995/07/09 Reimplemented using Kabelschacht method
[5] v1.0g 1995/08/16 Removed `\@whilenoop`
[6] v1.0g 1995/08/16 Made defs long

\@iwhilesw

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

\@nnil

A quark to mark delimited macros.

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

\@empty

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

\@fornoop

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

\@for{⟨:⟩}{⟨=⟩}{⟨*comma list*⟩}

Note that this is a macro delimited with `:=` and `\do`. After expanding `#2` it handles control to `\@forloop` which takes over control. If you wandering why `:=`, this was the `becomes` symbol used in Pascal and people were familiar with the convention.

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

\@forloop

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

\@iforloop

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

\@tfor {⟨*next token*⟩}{⟨*list*⟩}{⟨*body code*⟩}

The looping structure iterates over each token in a list. It uses the Kabelschacht method. Note that you may have to expand the list if in a macro, using an appropriate number of `\expandafter` commands.

Example 1.1 tfor

```
\makeatletter
\def\alist{ABCDE FGH}
\expandafter\@tfor\expandafter\next%
 \expandafter:\expandafter=\alist\do{%
  [\next]\tikz[inner sep=0pt, outer sep=0pt]\node (X)
      {\&};%
}
\makeatother
```

[A]&[B]&[C]&[D]&[E]&[F]&[G]&[H]&

```
\def\@tfor#1:={\@tf@r#1 }
```

\@tf@r

```
\long\def\@tf@r#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
    \@tforloop#2\@nil\@nil\@@#1{#3}\fi}

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

Note that it ignores any spaces in the list

(*End definition for* `\@tf@r`.)

\@break@tfor

Break out of a **\@tfor** loop. This should be called *inside* the scope of an **\if**. See \@iffileonpath for an example.

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

\@removeelement

Removes an element from a comma-separated list and puts it into a control sequence, called as **\@removeelement**{⟨*element*⟩}{⟨*list*⟩}{⟨*cs*⟩}. Due to the implementation method the ⟨*element*⟩ is not allowed to contain braces.

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

h a n g

7

Error Handling (file lterror.dtx)

This Chapter defines LaTeX 2_ε's error commands. Most of the error messages are defined here, although now and then some error messages are defined in other `.dtx` files. Warnings, fatal errors and help are amongst the techniques discussed.

21.00002
16
-2.5
-2.1
0.87865

7.1 Error handling

This section defines LaTeX 2_ε's error commands. Most of the error messages are defined here, although now and then some error messages are defined in other `.dtx` files.

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.

```
\expandafter\let\csname ver@autoerr.sty\endcsname\fmtversion
```

7.1.1 General commands

`\MessageBreak`

This command prints a new-line inside a message, followed by a continuation line beginning with `\@msg@continuation`. Normally it is defined to be `\relax`, but inside messages, it is let to **??**→P.?? for example in the definition of **??**→P.??.

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

`\GenericInfo`

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

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

`\GenericWarning`

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

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

`\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.

```
\def\GenericError{\@autoerr\GenericError}

\bgroup
\lccode`\@=`\ %
\lccode`\~=`\ %
\lccode`\}=`\ %
\lccode`\{=`\ %
\lccode`\T=`\T%
\lccode`\H=`\H%
\catcode`\ =11\relax%
\lowercase{%
\egroup%
```

Unfortunately TEX versions older than 3.141 have a bug which means that `^^J` does not force a linebreak in `\message` and `\errmessage` commands. So for these old TEX's we use `\typeout` to produce the message, and then have an empty `\errmessage` command. This causes an extra line of the form

```
! .
```

To appear on the terminal, but if you do not like it, you can always upgrade your TEX! In order for your format to use this version, you must define the macro `\@TeXversion` to be the version number, e.g., 3.14 of the underlying TEX. See the comments in `ltdircheck.dtx`.

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

First the 'standard case'.

```
\DeclareRobustCommand{\GenericError}[4]{%
\begingroup%
\immediate\write\@unused{}%
\def\MessageBreak{^^J}%
\set@display@protect%
\edef%
%    %<-------------------do not delete this ⤸
       ⤹space!------------------->%
\@err@ ⤸
       ⤹                                                          ⤸
       ⤹%
{{#4}}%
\errhelp
%    %<-------------------do not delete this ⤸
       ⤹space!------------------->%
\@err@ ⤸
       ⤹                                                          ⤸
       ⤹%
\let
%    %<-------------------do not delete this ⤸
       ⤹space!------------------->%
\@err@ ⤸
       ⤹                                                          ⤸
       ⤹%
\@empty
\def\MessageBreak{^^J#1}%
```

```
\def~{\errmessage{%
#2.^^J^^J%
#3^^J%
Type  H <return>  for immediate help%
%    %<-------------------do not delete this space!------------------->%
\@err@ ↲
        ↳                                                                ↲
        ↳%
}}%
~%
\endgroup}%
\else%
```

Secondly the version for old TEX's.

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

7.1.2 Package and Class error messages

`\PackageWarningNoLine`

`\PackageInfo`

`\ClassError`

`\ClassWarning`

`\ClassWarningNoLine`

`\ClassInfo`

These commands are intended for use by package and class writers, to give information to authors. The syntax is:

```
\PackageError{⟨package⟩}{⟨error⟩}{⟨help⟩}
\PackageWarning{⟨package⟩}{⟨warning⟩}
\PackageWarningNoLine{⟨package⟩}{⟨warning⟩}
\PackageInfo{⟨package⟩}{⟨info⟩}
```

and similarly for classes. The `Error` commands print the ⟨*error*⟩ message, and present the interactive prompt; if the author types `h`, then the ⟨*help*⟩ information is displayed. The `Warning` commands produce a warning but do not present the interactive prompt. The `WarningNoLine` commands do the same, but don't print the input line number. The `Info` commands write the message to the `log` file. Within the messages, the command `\MessageBreak` can be used to break a line, `\protect` can be used to protect command names, and `\space` is a space, for example:

```
\newcommand{\foo}{FOO}
\PackageWarning{ethel}{%
   Your hovercraft is full of eels,\MessageBreak
   and \protect\foo\space is \foo}
```

produces:

```
Package ethel warning: Your hovercraft is full of eels,
(ethel)                and \foo is FOO on input line 54.
```

The definition of the commands are fairly straight forward. They all use the `\GenericError` command to display the error on the terminal. The new expl3 module for LaTeX 3e, has expanded on this work tremendously, so that there is no real need for using these macros any longer.

`\PackageError` {⟨*package*⟩} {⟨*error*⟩} {⟨*help*⟩}

```
\gdef\PackageError#1#2#3{%
   \GenericError{%
      (#1)\@spaces\@spaces\@spaces\@spaces
   }{%
      Package #1 Error: #2%
   }{%
      See the #1 package documentation for explanation.%
   }{#3}%
}
```

\PackageWarning ⟨*package*⟩ ⟨*warning*⟩

```
\def\PackageWarning#1#2{%
   \GenericWarning{%
      (#1)\@spaces\@spaces\@spaces\@spaces
   }{%
      Package #1 Warning: #2%
   }%
}
\def\PackageWarningNoLine#1#2{%
   \PackageWarning{#1}{#2\@gobble}%
}
\def\PackageInfo#1#2{%
   \GenericInfo{%
      (#1) \@spaces\@spaces\@spaces
   }{%
      Package #1 Info: #2%
   }%
}
```

\ClassError

```
\gdef\ClassError#1#2#3{%
   \GenericError{%
      (#1) \space\@spaces\@spaces\@spaces
   }{%
      Class #1 Error: #2%
   }{%
      See the #1 class documentation for explanation.%
   }{#3}%
}
```

(*End definition for* `\ClassError`.)

```
\def\ClassWarning#1#2{%
   \GenericWarning{%
      (#1) \space\@spaces\@spaces\@spaces
   }{%
      Class #1 Warning: #2%
   }%
}
\def\ClassWarningNoLine#1#2{%
   \ClassWarning{#1}{#2\@gobble}%
}
\def\ClassInfo#1#2{%
```

```
   \GenericInfo{%
      (#1) \space\space\@spaces\@spaces
   }{%
      Class #1 Info: #2%
   }%
}
```

Some errors are marked to be generated only in the LaTeX 2e kernel. This is where we see the reference to the LaTeX Manual or the LaTeX Companion for help is mentioned.

\@latex@error

\@latex@warning

\@latex@warning@no@line

\@latex@info

\@latex@info@no@line

```
\gdef\@latex@error#1#2{%
   \GenericError{%
      \space\space\space\@spaces\@spaces\@spaces
   }{%
      LaTeX Error: #1%
   }{%
      See the LaTeX manual or LaTeX Companion for explanation.%
   }{#2}%
}
```

```
\def\@latex@warning#1{%
   \GenericWarning{%
      \space\space\space\@spaces\@spaces\@spaces
   }{%
      LaTeX Warning: #1%
   }%
}
```

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

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

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

`\@font@warning` and `\@font@info` are defined later since they have to be redefined by the `tracefnt` package.

```
\def\@font@warning#1{%
   \GenericWarning{%
```

```
        {(font)\@spaces\@spaces}%
        {Font Warning: #1}%
    }
  \def\@font@info#1{%
     \GenericInfo{%
        (font)\space\@spaces
     }{%
        Font Info: #1%
     }%
    }
```

`\c@errorcontextlines`

`\errorcontextlines` as a LaTeX counter, so that it may be be manipulated with `\setcounter` (once it is defined :-)

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

`\on@line`

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

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

`\@warning`

`\@@warning`

`\@latexerr`

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

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

`\@spaces`

Four spaces.

```
\def\@spaces{\space\space\space\space}
```

7.1.3 Specific errors

The four commands `\@eha`, `\@ehb`, `\@ehc`, and `\@ehd`, expand to specific errors. The more common error help messages are defined by macros named as a series of `\@eh`⟨*letter*⟩. Obviously short for errorhelp

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

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

```
%<*autoload>
\let\@eha\@empty\let\@ehb\@empty\let\@ehc\@empty\let\@ehd\@empty
%</autoload>
```

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

\@autoerr

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

```
<autoload>\def\@autoerr{\protect\@autoload{err}\protect}
```

\@notdefinable

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

```
\gdef\@notdefinable{%
```

\@nolnerr

Generated by **\newline** and **** when called in vertical mode.

```
\gdef\@nolnerr{%
<!autoload>  \@latex@error{There's no line here to end}\@eha}
<autoload>  \@autoerr\@nolnerr}
```

\@nocounterr

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

\@nocnterr

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

```
\gdef\@nocounterr#1{%
<!autoload>  \@latex@error{No counter '#1' defined}\@eha}
<autoload>  \@autoerr\@nocounterr}
\gdef\@nocnterr{\@nocounterr?}
```

\@ctrerr

Called when trying to print the value of a counter numbered by letters that's greater than 26.

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

\@nodocument

Error produced if paragraphs are typeset in the preamble.

```
<!def>\gdef\@nodocument{%
<!def>  \@latex@error{Missing \protect\begin{document}}\@ehd}
```

\@badend

Called by **\end** that doesn't match its **\begin**. RmS 1992/08/24: added code to **\@badend** to display position of non-matching **\begin**. FMi 1993/01/14: missing space added.

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

\@badmath

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

```
\gdef\@badmath{%
<!autoload>  \@latex@error{Bad math environment \protect\delimiter}\@eha}
<autoload>  \@autoerr\@badmath}
```

\@toodeep

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

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

\@badpoptabs

Called by **\endtabbing** when not enough **\poptabs** have occurred, or by **\poptabs** when too many have occurred.

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

\@badtab

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

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

\@preamerr

This error is special: it appears in places where we normally have to **\protect** expansions. However, to prevent a protection of the error message itself (which would result in the message getting printed not issued on the terminal) we need to locally reset **\protect** to **\relax**.

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

\@badlinearg

Occurs in **\line** and **\vector** command when a bad slope argument is encountered.

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

\@parmoderr

Occurs in a float environment or a **\marginpar** when encountered in inner vertical mode.

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

\@fltovf

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

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

\@latexbug

Occurs in output routine. This is bad news.

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

\@badcrerr

This error was removed and replaced by **\@nolnerr**.

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

\@noitemerr

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

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

\@notprerr

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

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

\@inmatherr

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

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

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

Environment --- undefined

Issued by **\begin** for undefined environment.

tab overflow

Occurs in **\= when** maximum number of tabs exceeded.

\< in mid line

Occurs in **\<** when it appears in middle of line.

Float(s) lost

In output routine, caused by a float environment or **\marginpar** occurring in inner vertical mode.

h a n g

8

ltspace.dtx

This Chapter defines LaTeX 2_ε's error commands. Most of the error messages are defined here, although now and then some error messages are defined in other `.dtx` files. Warnings, fatal errors and help are amongst the techniques discussed.

21.00002
16
-2.5
-2.1
0.87865

8.1 Spacing

This section of the LaTeX kernel deals with spacing, line and page-breaking.

8.1.1 User Commands

`\nopagebreak`{⟨*0,...,4*⟩}

The command**??** → P. **??** [⟨*i*⟩] : ⟨*i*⟩ = 0,...,4 attempts to stop the page breaking at this point, by inserting an appropriate amount of penalty.

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

Interestingly the definitions of the penalties is left to the class files such as book. How are `\@lowpenalty` etc. defined? Their definition is found in the class file, although the counters are defined in the kernel source.:

```
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
```

As Frank Mittelbach noted http://tex.stackexchange.com/questions/83350/nopagebreak-and-penalty: `\nopagebreak` is a high-level LaTeX command that inspects its suroundings (to some extend) and does different things depending on whether it is used in vertical or in horizontal mode.

If you are in the middle of a paragraph, then saying `\penalty` 10000 is inserting a penalty that directs the linebreaking, but not one that affects the page breaking. In contrast `\nopagebreak` would insert more or less `\vadjust{\penalty 10000}` and it would do some additional magic to ensure that you don't end up with two spaces in that place.

So bottom line, as a user you shouldn't use `\penalty` at all, there are good reasons why LaTeX provides higher-level user commands. As a developer of a package, you may use penalty if you know what situation are in and of course using `\penalty` directly removes a tiny bit of overhead.

`\pagebreak`[⟨*i*⟩]

`\pagebreak` is the same as `\nopagebreak` except negatives of its penalty.

`\linebreak`[⟨*i*⟩]

analog of the above

`\nolinebreak`[⟨*i*⟩]

analog of the above

`\samepage*`

Inhibits page breaking most places by setting the following penalties to 10000:
`\interlinepenalty`
`\postdisplaypenalty`
`\interdisplaylinepenalty`
`\@beginparpenalty`
`\@endparpenalty`
`\@itempenalty`
`\@secpenalty`
`\interfootnotelinepenalty`

Example 1.1 Example

```
\bgroup
\samepage

\the\interlinepenalty

\the\postdisplaypenalty

\the\interfootnotelinepenalty
\egroup

\the\interlinepenalty
```

```
10000
10000
10000
0
```

8.1.2 Vertical spacing commands

The vertical spacing commands are similar to the horizontal commands. The plain TEX commands **??**→P.**??**, **??**→P.**??** and **??**→P.**??** are supported by LATEX, however they are redefined in terms of `\vspace` instead of `\vskip`.

`\smallskip`

`\medskip`

`\bigskip`

`\vspace`⟨*⟩{⟨*dim*⟩}

The commands `\vspace` and `\vspace*` add a fixed amount of vertical spacing.

`\addvspace`{⟨*skip*⟩}

Extra vertical space is added by the comand `\addvspace`. It adds a vertical amount of skip of ⟨*skip*⟩ to the document. The command can only be used in vertical mode. The **??**[→P.??] command does not add vertical space if `minipage` is true.

8.1.3 Horizontal spacing

`\hspace`

`\hspace*`

The commands `\hspace`{⟨*skip*⟩} and `\hspace*`{⟨*skip*⟩} are robust commands that add a horizontal skip. The star version of the command should be used at the beginning of a line.

`\nobreakspace`

The no breaking space ~ The **??**[→P.??] command is a robust command that produces a horizontal space at which, in paragraph mode, a line-break is not possible. The ĩs defined as active to expand to **??**[→P.??].

When `\.` is used in paragraph mode it produces a `\thinspace`. The command `\@` when placed before a '.' makes it a sentence-ending period. It also does the right thing after other punctuation marks as well.

The command `\thinspace` adds a .16667em kern, whereas the `\negthinspace` adds an equal negative amount of kern. An `\enspace` is a .5em kern.

`\thinspace`

Adds a .16667em kern.

`\negthinspace`

`\enspace`

8.1.4 Obsolete Commands

`\obeycr` : defines <CR> == `\\\relax`
`\restorecr` : restores <CR> to its usual meaning.

8.1.5 The code

The code follows the sequence and commentary of the original source to a large extend. I have added additional commentary where necessary to provide additional explanatory comments and to cross-reference code to algorithms and other sections.

```
\message{spacing,}
```

In the definition of **??**[→P.??] and **??**[→P.??] observe the use of **??**[→P.??]. This macro is defined in **ltdefns.dtx** and encapsulates the most common call to **??**[→P.??]. It takes two arguments. There is also a robust version available **??**[→P.??].

\pagebreak

The control sequence **\pagebreak** inserts penalties to force a page break.

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

\nopagebreak

The opposite of **\pagebreak**

```

\def\nopagebreak{\@testopt\@no@pgbk4}
```

\@no@pgbk

This is the main internal macro for adding the penalties. Penalties are added after examining if the command has been issued in vertical or horizontal mode. The command **\@getpen** is defined on Line **??**.

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

\linebreak

\nolinebreak

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

\@no@lnbk

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

\samepage

The macro \samepage is a versatile macro that can be used in many different environments to set penalties, in such a way as to discourage page breaking. All penalties are set to \@M

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

\\

It appears to be safe to use \reserved@e and \reserved@f here (other reserved macros are somewhat disastrous).

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

\@normalcr

The internal definition of the 'normal' definition of \\.

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

\newline

The \newline provides syntactic sugar to a simple form of the \@normalcr definition of \\.

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

\@xnewline

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

\@newline

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

\@gnewline

The \nobreak added to prevent null lines when \\ ends an overfull line.

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

\@getpen

The command \@getpen is a helper command to allocate penalties. This has been abstracted to make it easier for classes to insert penalties that they predefine.

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

\if@nobreak

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

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

8.1.6 Space removing macros

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

Before giving the current definition we give an extended definition that is currently not used (because it doesnt work as advertised:-)

These are generalised hacks which attempt to do sensible things when 'invisible commands' appear in vmode too.

They need to cope with space in both hmode (plus spacefactor) and vmode, and also cope with breaks etc. In vmode this means ensuring that any following \addvspace, etc sees the correct glue in \lastskip.

In fact, these improved versions should be used for other cases of 'whatsits, thingies etc' which should be invisible. They are only for commands, not environments (see notes on \@Esphack).

BTW, anyone know why the standard hacks are surrounded by \ifmmode\else rather than simply \ifhmode?

And are there any cases where saving the spacefactor is essential? I have some extensions where it is, but it does not appear to be so in the standard uses.

\@savsk{⟨*skip*⟩}

\@savsfnumber

The registers \@savsk and \@savsf used to save the space factor and last skip.

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

\@bsphack

```
\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
}
```

\@bsphack

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

\@esphack

Companion to \@bsphack.

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

\@Esphack

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

8.1.7 Vertical spacing

LaTeX supports the plain TeX commands **\smallskip**, **\medskip** and **\bigskip**. However, it redefines them using **\vspace** instead of **\vskip**.

Extra vertical space is added by the command **\addvspace**{⟨*skip*⟩}, which adds a vertical skip of ⟨*skip*⟩ to the document. The sequence **\addvspace**{⟨*s1*⟩} **\addvspace**{⟨*s2*⟩} is equivalent to **\addvspace**{⟨*maximum of s1, s2*⟩}.

\addvspace should be used only in vertical mode, and gives an error if it's not. The **\addvspace** command does *not* add vertical space if **@minipage** is true. The minipage environment uses this to inhibit the addition of extra vertical space at the beginning.

Penalties are put into the vertical list with the **\addpenalty**{⟨*penalty*⟩} command. It works properly when **\addpenalty** and **\addvspace** commands are mixed.

The **@nobreak** switch is set true used when in vertical mode and no page break should occur. (Right now, it is used only by the section heading commands to inhibit page breaking after a heading.)

```
\addvspace{⟨skip⟩} == begin
  if vmode then
    if @minipage then
    else
      if \lastskip =0 then
        \vskip ⟨skip⟩
      else
        if \lastskip < ⟨skip⟩ then
          \vskip -\lastskip \vskip ⟨skip⟩
        else
          if SKIP < 0 and \lastskip >= 0 then
            \vskip -\lastskip
            \vskip \lastskip + ⟨skip⟩
          end
        end
      end
    end
  else
    Error message.
  end
end
```

Algorithm 1: The `\addvspace` command.

`\@xaddvskip`

Internal macro for `\vspace` handling the case that space has previously been added.

```
\def\@xaddvskip{%
  \ifdim\lastskip<\@tempskipb
    \vskip-\lastskip
    \vskip\@tempskipb
  \else
    \ifdim\@tempskipb<\z@
      \ifdim\lastskip<\z@
      \else
        \advance\@tempskipb\lastskip
        \vskip-\lastskip
        \vskip \@tempskipb
      \fi
    \fi
  \fi}
```

This is actually a good place to create an example of usage and to also revisit the `\lastskip` and `\unskip` commands.

Example 1.2 Unskip and lastskip

```
\bgroup
\newdimen\Tkern
\setbox1=\hbox{\kern1pt\hbox{B}\quad}
\unhbox1
\the\lastskip
\skip0=\lastskip \unskip
\setbox0=\lastbox
\the\lastskip
\Tkern=\lastkern \unkern
\setbox2=\hbox{\hskip\skip0\box0\kern\Tkern}
\egroup
```

B 9.0pt0.0pt

\addvspace

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

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

\addpenalty⟨*penalty*⟩

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

```
    \fi
  \else
    \@noitemerr
  \fi}
```

\vspace⟨*glue*⟩

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

\smallskip

```
 \def\smallskip{\vspace\smallskipamount}
```

\medskip

```
 \def\medskip{\vspace\medskipamount}
```

\bigskip

```
\def\bigskip{\vspace\bigskipamount}
```

\smallskipamount

\medskipamount

\bigskipamount

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

8.1.8 Horizontal space (and breaks)

\nobreakdashes

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

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

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

Setting the hyphen in a box and then unboxing it means that the normal penalty will not be added after it—and if the penalty is not there a break will not be taken (unless an explicit penalty or glue follows, thus the final `\nobreak`).

Note that even if it is not followed by a '-', it still leaves vmode and sets the spacefactor; so use it carefully!

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

\nobreakspace

This is a robust command that produces a horizontal space at which, in paragraph-mode, a line-break is not possible. We then define an active `~` to expand to it since this is the documented behaviour of `~`. One reason for introducing this is that some 8-bit input encodings have a slot for such a space and we do not want to use active characters as the LaTeX internal commands.

The braces in the definition of ~ are needed to ensure that a following space is preserved when reading to/from internal files. We need to keep **\@xobeysp** as it is widely used; so here it is let to the non-robust command **\nobreakspace** .

```
\DeclareRobustCommand{\nobreakspace}{%
   \leavevmode\nobreak\ }
\catcode `\~=13
\def~{\nobreakspace{}}
\expandafter\let\expandafter\@xobeysp\csname nobreakspace ↩
      ↪\endcsname
```

\,

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

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

\@

Placed before a '.', makes it a sentence-ending period. Does the right thing for other punctuation marks as well. Does this by setting spacefactor to 1000.

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

\hspace⟨*skip*⟩

The command **\hspace** is declared as robust, so that it can be used in moving argument. The unstarred version is simply an **\hskip**.

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

\@hspace⟨*skip*⟩

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

\@hspacer⟨*skip*⟩

The star version of the command is more complex. It starts with a zero width **\vrule**, followed by a **\nobreak**. It guards against a following unskip by a **\z@skip**. The latter is defined in ltplain.dtx as **\newskip\z@skip \z@skip=0pt plus0pt minus0pt**.

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

\fill

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

\stretch

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

\thinspace

\negthinspace

\enspace

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

\enskip

\quad

\qquad

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

\obeycr

\restorecr

A not ein the code, states that 'the following definitions will probably get deleted or moved to compatibility mode soon.', I guess they will be here for a few more decades.

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

h a n g

9

Logos

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed. These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002
16
-2.5
-2.1
0.87865

\TeX The TEX logo, adjusted so that a full stop after the logo counts as ending a sentence.

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

(*End definition for* \TeX.)

\LaTeX The LATEX logo. [1] [2] [3] [4] [5]

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

(*End definition for* \LaTeX.)

[6]

\LaTeXe The LATEX 2_ε logo as proposed by A-W designers. [7] [8] [9] [10] [11]

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

(*End definition for* \LaTeXe.)

[1] LaTeX2e 1993/11/24 Macro changed
[2] v1.00 1994/05/11 Use **\DeclareProtectedCommand**. ASAJ.
[3] v1.1d 1994/06/01 Add **\m@th** to force math size calculations
[4] v1.1e 1994/07/20 Save a few tokens
[5] v1.1h 1997/10/08 Simplify macro (force loading of suitable math fonts once).
[6] v1.1c 1994/05/26 Remove **\SLiTeX** logo
[7] LaTeX2e 1993/11/22 Macro added
[8] LaTeX2e 1993/12/16 Extended logo by DPC
[9] v1.00 1994/05/11 Use **\DeclareProtectedCommand**. ASAJ.
[10] v1.1e 1994/07/20 Save a few tokens
[11] v1.1f 1997/05/29 Added **\m@th** so that the LATEX 2_ε logo works with non-zero values of **\mathsurround**.

h a n g

10

File Handling

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed.
These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002
16
-2.5
-2.1
0.87865

The following user commands are defined in this part (k ltfiles.dtx):

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

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

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

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

`\input` {⟨*NAME*⟩}
The same as TeX's `\input`, except it allows optional braces around the file name. In LaTeX 2_ε, it also avoids the primitive 'missing file' error, if the file can not be found.

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

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

10.1 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 := True
    \@partlist := FILELIST
end
```

```
\include{⟨FILE⟩} == begin
    \clearpage
    if \@filesw = True then
        \immdiate\write\@mainaux\string\@input{⟨FILE.AUX⟩}
    end
end
```

`\@mainaux`

`\@partaux`

These two macros allocate write streams for the main auxiliary file and the part file.

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

Next the switches are defined as well as a newcount for `\@clubpenalty`. The latter still puzzles me as to why here.

`\if@filesw`

`\if@partsw`

```
\newif\if@filesw \@fileswtrue
\newif\if@partsw \@partswfalse
```

`\@clubpenalty`

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

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

`\document`

Cancel the `\begingroup` from `\begin`

```
 \def\document{\endgroup
```

If some options on `\documentclass` haven't been used by any package we will now give a warning since this is most certainly a misspelling.

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

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

```
\process@table
\let\glb@currsize\@empty  %% Force math initialization.
\normalsize
\everypar{}%
```

So that punctuation in headings is not disturbed by verbatim or other local changes to the space factor codes, save the document default here. This will be locally reset by the output routine. For special cases a class may want to define `\normalsfcodes` directly, in case that definition will be used. (This is an old bug, problem existed in LaTeX2.0x and plain TeX.)

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

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

```
\@noskipsecfalse
```

changesv1.1a1995/10/24 Removed refundefined switch

```
\let \@refundefined \relax
```

Just before disabling the preamble commands we execute the begin document hook which contains any code contributed by `\AtBeginDocument`.

Also disable the gathering of the file list, if no **\listfiles** has been issued. **\AtBeginDocument** is redefined at this point so that and such commands that get into the hook do not chase their tail...

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

Most of the following assignments will be done globally in case the user adds something like **\begin{multicols}** to the document hook, i.e. starts are group in **\begin{document}**. Since a value of exactly opt for **\topskip** causes **\twocolumn[]** to misbehave, we add this check, hoping that it will not cause any problems elsewhere.

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

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

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

Added disabling of **\@nodocument**

```
\global\let \@nodocument \relax
```

The next line is a pure safety measure in case a do list is ever expanded at the wrong place. In addition it will save a few tokens to get rid of the above definition.

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

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

```
\ignorespaces}
```

```
\@onlypreamble\document
```

\normalsfcodes

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

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

\nofiles

Set **\@fileswfalse** which suppresses the places where LaTeX makes **\immediate** writes. The **\makeindex** and **\makeglossary** are disabled. **\protected@write** is redefined not to write to the file specified,

but rather to write a blank line to the log file. This ensures that a ⟨*whatsit*⟩ node is still created, and so spacing is not affected by the **\nofiles** command; to ensure this more generally, the **\if@nobreak** test is needed.

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

\protected@write

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

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

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

\includeonly

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

\include

In the definition of **\include**, **\def\reserved@b** changed to **\edef\reserved@b** to be consistent with the **\edef** in **\includeonly**. (Suggested by Rainer Schöpf & Frank Mittelbach. Change made 20 Jul 88.)

Changed definition of **\include** to allow space at end of file name — otherwise, typing **\include{foo }** would cause LaTeX to overwrite **foo.tex**. Change made 24 May 89, suggested by Rainer Schöpf and Frank Mittelbach

Made **\include** check for being used inside an **\include**'d file, as this will not work and cause surprising results.

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

\@include

The definition of **\@include** is fairly straightforward less one puzzle, why would the **\clearpage** be issued and thus have a behaviour which is totally different from **\input**. As explained by egreg on TEX.SX this was to correct the references. David Carlisle added that floats can also get passed none included files, which is also an issue. [1]. This is a very subtle error and easily missed. David Carlisle added that: "...as a historical note it wasn't a "team" it has always done this so this is Leslie's doing before we got involved". Frank provided a more lengthy explanation which I quote:

> The purpose of the **\include** mechanism is to allow for partially compiling the document when making a modest amount of changes without the need to recompile the full document and still get cross-references etc correctly resolved (even to parts outside the current part under the knife).
>
> For this to work the part or parts that are included have to be selfcontained in the sense that changes to it do not automatically (and always) render the formatting of other parts not included invalid. To make this possible (at least for smaller changes) the following prerequisites are needed:
>
> The mechanism need to ensure that small changes in text length in the part being compiles doesn't result in formatting changes in other parts not compiled Floats in the parts compiled need to be placed in the parts compiled If either of the above points is not valid then using **\include** would (nearly) always result in invalid documents whereas in the current scheme you can all include parts individually and still arrive and maintain a valid document. (Just for the record, when we produced the first edition of the LaTeX Companion it took ages to compile this book and even the second edition took about 30 minutes in 2004 to compile all examples + all pages and rerun the whole book (I think 5 times) to get all cross-references resolved successfully. So building out things chapter by chapter was essential and even then compiling took long :-)
>
> Of course the first point will be invalid the moment a lot of text is added or removed such as the resulting document get one or more pages longer or shorter.
>
> So using page breaks at the boundaries of the included parts is simply necessary to make the mechanism worth while in the first place and using **\clearpage** is needed to to ensure that the floats stay within the included part and not move in or out.
>
> @egreg already gave the additional explanation that the mechanism to write aux file data works only on **\shipout** so that it wouldn't be possible (or at least not easy) to ensure that such things like cross-references or data for table of contents aren't lost. Technically one could think of a possibility to manage this, but using more than one aux file per include, but that wouldn't resolve the point above.

[1] http://tex.stackexchange.com/questions/108268/redefining-include

Finally, this is not something the LaTeX Project team invented, it goes back to Leslie Lamport's original design and was in existance since LaTeX 2.08 (at least) so before 1986.

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

If the file is not included, reset **\deadcycles**, so that a long list of non-included files does not generate an 'Output loop' error.

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

\@writeckpt

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

\@wckptelt

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

\@setckpt

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

\@charlb

\@charrb

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

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

10.2 Safe Input Macros

\IfFileExists

The predicate \IfFileExists tries to open the file by opening a stream \@inputcheck.

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

\@iffileonpath

If the file is not found by \openin, and \input@path is defined, look in all the directories specified in \input@path. Note the use of \@tfor and if found how to break using \@break@tfor.

```
\long\def\@iffileonpath#1{%
  \let\reserved@a\@secondoftwo
  \expandafter\@tfor\expandafter\reserved@b\expandafter
              :\expandafter=\input@path\do{%
    \openin\@inputcheck\reserved@b#1 %
    \ifeof\@inputcheck\else
```

```
      \edef\@filef@und{\reserved@b#1 }%
      \let\reserved@a\@firstoftwo%
      \closein\@inputcheck
      \@break@tfor
    \fi}%
  \reserved@a}
```

\InputIfFileExists{⟨*filename*⟩}{⟨*true code*⟩}{⟨*false code*⟩}

Now define **\InputIfFileExists** to input **#1** if it seems to exist. Immediately prior to the input, **#2** is executed. If the file **#1** does not exist, execute '**#3**'.

```
\long\def \InputIfFileExists#1#2{%
  \IfFileExists{#1}%
    {#2\@addtofilelist{#1}\@@input \@filef@und}}
```

\input{⟨*filename*⟩}

Input a file: if the argument is given in braces use safe input macros, otherwise use TEX's primitive **\input** command (which is called **\@@input** in LATEX).

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

\@iinput

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

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

\@input

Define **\@input** in terms of **\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 **.bbl** probably should be listed and thus should be implemented not by **\@input**.

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

\@input@

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

```
\def\@input@#1{\InputIfFileExists{#1}{}{\typeout{No file #1.}}}
```

\@missingfileerror

This 'error' command avoids TEX's primitive missing file loop. Missing file error. Prompt for a new filename, offering a default extension.

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

\@obsoletefile

For compatibility with LATEX 2.09 document styles, we distribute files called `article.sty`, `book.sty`, `report.sty`, `slides.sty` and `letter.sty`. These use the command `\@obsoletefile`, which produces a warning message.

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

10.3 Listing files

\@filelist

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

```
\let\@filelist\@gobble
```

\@addtofilelist

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

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

\listfiles

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

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

\@dofilelist

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

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

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

```
\@onlypreamble\listfiles
```

Set **\@dofilelist** to an initial value of **\relax**.

```
\let\@dofilelist\relax
```

Hope you are still with me and that the explanations have been adequate. I still think some design decisions should not be carried over to LaTeX3, such as the **\clearpage** before an **\include** file.

h a n g

11

ltoutenc

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed. These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002

16

-2.5

-2.1

0.87865

11.1 Background

The Latin Modern fonts, created as a successor to Knuth's Computer Modern fonts contain some 50,000 glyphs. The fonts were authored by Boguslaw Jackowski and Janusz M. Nowacki.[1] There usage was described in an article in **Robertson2007**.[2] Modern engines such as XeTeX and LaTeX do not have any difficulty in using Open Type fonts natively with thousands of glyphs, however using fonts while TeX is frozen is a challenge. There are two major challenges that the LaTeX Team and the community historically overcame.

1. Providing mechanisms to map 256 symbol fonts to encodings and hence permit combinations to be used.
2. Provide a comprehensive set of markup commands to handle thousands of symbols.

Representation as "font -encoding-specific" commands

The other way to represent characters internally in TeX (and this covers the majority of characters) is with special LaTeX commands (or command sequences) that remain unexpanded when written to a file or when placed into a moving argument. These special commands are sometimes referred to as *font-encoding-specific commands* because their meaning depends on the font encoding current when LaTeX is ready to typeset them. Such commands are declared using special declarations, as discussed below. They usually require individual definitions for each font encoding. If no definition exists for the current encoding, either a default is used (if available) or an error message is presented to the user.

Technically, when the font encoding is changed at some point in the document, the definitions of the encoding-specific commands do not change immediately, as that would mean changing a large number of commands on the spot. Instead, these commands have been implemented in a such way that they notice, once they are used, if their current definition is no longer suitable for the font encoding in force. In such a case they call upon their counterparts in the current font encoding to do the actual work.

The set of "font-encoding-specific commands" is not fixed, but rather implicitly defined to be the union of all commands defined for individual font encodings. Thus, by adding new font encodings to LaTeX, new "font-encoding-specific commands" might emerge.

11.2 Output encodings

Output encodings define the mapping from the LICR to the glyphs (or constructs built from glyphs) available in the fonts used for typesetting. These mappings are referenced inside TeX by two or three-letter names (e.g., OT1 and

[1] **boguslaw2003**.
[2] **Robertson2007**.

T3). We say that a certain font is in a certain encoding if the mapping corresponds to the positions of the glyphs in the font in question. So what are the exact components of such a mapping?

Characters internally represented by ASCII characters are simply passed on to the font. In other words, TEX uses the ASCII code to select a glyph from the current font. For example, the character "A" with ASCII code 65 will result in typesetting the glyph in position 65 in the current font. This is why TEX requires that fonts for text contain all such ASCII letters in their ASCII code positions, as there is no way to interact with this basic TEX mechanism (other than to disable it and do everything "manually"). Thus, for visible ASCII, a one-to-one mapping is implicitly present in all output encodings.

Characters internally represented as sequences of ASCII characters (e.g., "–"), are handled as follows: when the current font is first loaded, TEX is informed that the font contains a number of so-called ligature programs. These define certain character sequences that are not to be typeset directly but rather to be replaced by some other glyphs from the font (the exact position of each replacement glyph is font dependent and not important otherwise). For example, when TEX sees "" in the input (Le., ASCII code 45 twice), a ligature program might direct it to use the glyph in position 123 instead (which then would hold the glyph "–"). Again, no interaction with this mechanism is possible. Some such ligatures are present for purely aesthetic reasons and mayor may not be available in certain fonts (e.g., ff generating "ff" rather than IIff"). Others are supposed to be implemented for a certain encoding (e.g., "—" producing an —).

Nevertheless, the bulk of the internal character representation consists of "font-encoding-specific" commands. They are mapped using the declarations described below. All declarations have the same structure in their first two arguments: the font-encoding-specific command (or the first component of it, if it is a command sequence), followed by the name of the encoding. Any remaining arguments will depend on the type of declaration.

Thus, an encoding `XYZ` is defined by a number of declarations all having the name `XYZ` as their second argument. Of course, to be of any use, some fonts must be encoded in that encoding. In fact, the development of font encodings is normally done the other way around-namely, someone starts with an existing font and then provides appropriate declarations for using it. This collection of declarations is then given a suitable name, such as `0T1`. In the next section, we will take the font ecrml000, shown in Table 7.32 on the facing page, whose font encoding is called T1 in TEX, and build appropriate declarations to access the glyphs from a font encoded in this way. The blue characters in this table are those that have to be present in the same positions in every text encoding, as they are transparently passed through TEX.

11.2.1 Output encoding files

Like input encoding files, output encoding files are identified by the extension **`.def`**. However, the base name of the file is slightly more structured: the name of the encoding in lowercase letters, followed by the letters **`enc`** (e.g., **`t1enc.def`** for the **`Tl`** encoding).

Such files should contain only the declarations described in the current

section. As output encoding files might be read several times by TEX, it is particularly important to adhere to this rule strictly and to refrain from using, for example, `\newcommand`, which prevents reading such a file multiple times! For identification purposes an output encoding file should start with a `\ProvideFile` declaration describing the nature of the file. For example:

```
\ProvidesFile{tlenc.def}[2001/06/05 vl.94 Standard LaTeX file]
```

To be able to declare any encoding-specific commands for a particular encoding, we first have to make this encoding known to TEX. This is achieved via the **??**[→ P. ??] declaration. At this point it is also useful to declare the default substitution rules for the encoding with the help of the command **??**[→ P. ??]; both declarations are described in detail in Section 7.10.5 starting on page 430.

11.3 Font encodings

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

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

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

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

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

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

The syntax for declaring encoding-specific commands is:

`\DeclareTextCommand{` ⟨*command*⟩`}{` ⟨*encoding*⟩`}`
`[` ⟨*number*⟩`][` ⟨*default*⟩`]{` ⟨*commands*⟩`}`

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

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

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

\ProvideTextCommand{⟨*command*⟩}{⟨*encoding*⟩}
[⟨*number*⟩][⟨*default*⟩]{⟨*commands*⟩}

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

\DeclareTextSymbol{⟨*command*⟩}{⟨*encoding*⟩}{⟨*slot*⟩}

This command defines a text symbol, with a particular slot in that encoding. The commands: [3]

```
\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. [4]

\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}
```

[5] 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}
```

[3] v1.9h 1997/12/17 Example corrected, braces removed.
[4] v1.8c 1996/10/27 Corrected syntax descriptions
[5] v1.8e 1996/11/23 Corrected description

[6] 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 **??** → P. ??:

`\UseTextSymbol{`⟨*encoding*⟩`}{`⟨*command*⟩`}`

[7] 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}}
```

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

`\DeclareTextCommandDefault{`⟨*command*⟩`}{`⟨*definition*⟩`}`

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

```
\DeclareTextCommandDefault{\textonequarter}{\ensuremath {\frac14}}
```

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

`\ProvideTextCommandDefault{`⟨*command*⟩`}{`⟨*definition*⟩`}`

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

`\DeclareTextSymbolDefault{`⟨*command*⟩`}`⟨*encoding*⟩`}`
`\DeclareTextAccentDefault{`⟨*command*⟩`}`⟨*encoding*⟩`}`

are short for:

[6] v1.8e 1996/11/23 Corrected description
[7] v1.9e 1997/08/05 Corrected order of arguments in `\UseTextSymbol` example.
[8] v1.8e 1996/11/23 Extended description

```
\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**.

11.3.1 Removing encoding-specific commands

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

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

is not enough since if typesetting in OT1 LaTeX will still find the encoding specific-definition for OT1 and therefore ignore the new default. Therefore to ensure that in this case the TS1 version is used we have to remove the OT1 declaration:

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

Since the $ sign is a proper glyph in the T1 encoding there is no point removing its definition and forcing LaTeX to pick up the TS1 version if typesetting in this encoding. However, assume you want to use the variant dollar sign, i.e., $ for your dollars. In that case you have to get rid of the T1 declaration as well, e.g., the following would do that for you:

```
\UndeclareTextCommand{\textdollar}{OT1}
\UndeclareTextCommand{\textdollar} {T1}
\DeclareTextCommandDefault{\textdollar}
                        {\UseTextSymbol{TS1}\textdollaroldstyle}
```

11.3.2 The order of declarations

[9]

[9] v1.90 1998/03/20 Documentation added about order of decls

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.

11.3.3 Docstrip modules

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

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

10 11 12 13 14 15 16 17 18

[10] 1.0d 1993/07/17 changed `\catcoding` @

[11] 1.0f 1993/08/13 Protected against active @ sign.

[12] 1.0g 1993/08/16 Needs space after `\string`

[13] 1.1 1993/12/07 Protected all special characters with `\string`.

[14] 1.3 1993/12/17 Removed the catcode hackery, since the file is only read as a package in the preamble, and removed all the messages on the screen, which just confuse users. Replaced them by the appropriate `\ProvidesPackage` commands. Added XXXenc.

[15] 1.3b 1993/12/18 Fixed typos with `\ProvidesPackage` lines. Added the `\NeedsTeXFormat` line. Added the last argument to `\DeclareEncoding`. Moved the use of the encodings to after their declaration.

[16] 1.4a 1994/04/29 Removed Rokicki's OT1 variant encoding. Moved the driver to the top.

[17] 1.5a 1994/05/11 Made T1 and OT1 generate packages rather than def files. Renamed the 'package' module to 'teststy'.

[18] 1.5d 1994/05/14 Moved the driver to the top.

11.3.4 Definitions for the kernel

11.3.40 Declaration commands

This section contains definitions for commands such as accents which depend on the current encoding. These commands will usually be kept in `.def` files, for example `ot1enc.def` contains the definitions for the `OT1` encoding.

```
%<*2ekernel>
\message{font encodings,}
```

`\DeclareTextCommand`

`\ProvideTextCommand`

`\DeclareTextSymbol`

`\@dec@text@cmd`

`\chardef@text@cmd`

`\@changed@cmd`

`\@changed@x`

`\TextSymbolUnavailable`

`\@inmathwarn`

If you say:

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

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

```
\def\DeclareTextCommand{%
   \@dec@text@cmd\newcommand}
\def\ProvideTextCommand{%
   \@dec@text@cmd\providecommand}
\def\@dec@text@cmd#1#2#3{%
   \expandafter\def\expandafter#2%
      \expandafter{%
         \csname#3-cmd\expandafter\endcsname
         \expandafter#2%
         \csname#3\string#2\endcsname
      }%
   \let\@ifdefinable\@rc@ifdefinable
   \expandafter#1\csname#3\string#2\endcsname}
```

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

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

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

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

```
\@onlypreamble\DeclareTextCommand
\@onlypreamble\DeclareTextSymbol
```

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

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

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

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

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

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

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

```
\gdef\TextSymbolUnavailable#1{%
   \@latex@error{%
      Command \protect#1 unavailable in encoding \cf@encoding%
   }\@eha}
```

The command `\@inmathwarn` produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't call `\relax` before the `\ifmmode`. This means that it is possible for the warning to fail to be issued at the beginning of a row of an halign whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a `\relax` would break ligatures and kerning between text symbols.

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

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

`\DeclareTextCommandDefault`

These define commands with encoding `?`. Note that `\DeclareTextCommandDefault` can only be used in the preamble, but that the `\Provide` version is allowed in inputenc `.def` files, so is allowed anywhere.

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

\ProvideTextCommandDefault

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

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

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

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

\DeclareTextAccent

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

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

```
\@onlypreamble\DeclareTextAccent
```

\add@accent

To save space this code is shared between all text accents that are set using the `\accent` primitive. The argument is pre-set in a box so that any font loading that is needed is already done within the box. This is needed because font-loading involves grouping and that would prevent the accent mechanism from working so that the accent would not be positioned over the argument. Declarations that change the font should be allowed (only low-level ones are at present) inside the argument of an accent command, but not size changes, as they involve `\setbox` operations which also inhibit the mechanism of the `\accent` primitive. Note that the whole process is within a group. For a detailed discussion of this reimplementation and its deficiencies, see pr/3160.

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

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

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

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

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

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

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

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

\@text@composite
\@text@composite@x
\@strip@args

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

```
\@onlypreamble\DeclareTextCompositeCommand
```

This all works because:

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

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

Note that `\@text@composite` grabs the first token of the argument and puts just that in the csname. This is so that `\'{\textit{e}}` will work—it checks whether `\\T1\'-\textit` is defined (which presumably it isn't) and so expands to `{\accent 1 \textit{e}}`.

This trick won't always work, for example `\'{{\itshape e}}` will expand to (with spaces added for clarity):

```
\csname \string \T1\' - \string {\itshape e} \@empty \endcsname
```

which will die pretty horribly. Unfortunately there's not much can be done about this if we're going to use `\csname` lookups as a fast way of accessing composites.

This has an unfortunate 'misfeature' though, which is that in the T1 encoding, `\'{aa}` produces á. This is not the expected behaviour, and should perhaps be fixed if the fix doesn't affect performance too badly.

Finally, it's worth noting that the `\@empty` is used in `\@text@composite` so that accents will work even when the argument is empty. If you say `\'{}` then this looks up `\\T1\'-\@empty`, which ought to be `\relax`, and so all is well. If we didn't include the `\@empty`, then `\'{}` would expand to:

```
\csname \string \T1\' - \string \endcsname
```

so the `\endcsname` would be `\string`'ed and the whole of the rest of the document would be put inside the `\csname`. This would not be good.

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

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

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

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

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

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

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

```
\@onlypreamble\DeclareTextComposite
```

(*End definition for* `\@text@composite`, `\@text@composite@x`, *and* `\@strip@args`.)

\UseTextSymbol

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.

\UseTextAccent

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

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

```
    \let\hmode@start@before@group\@firstofone
    \let\@curr@enc\cf@encoding
    \@use@text@encoding{#1}%
    #2{\@use@text@encoding\@curr@enc#3}%
   }}
```

```
\def\UseTextSymbol#1#2{%
        \hmode@start@before@group
        {%
           \def\@wrong@font@char{\MessageBreak
              for \noexpand\symbol`\string#2'}%
           \@use@text@encoding{#1}%
           #2%
        }%
     }
\def\@use@text@encoding#1{%
   \edef\f@encoding{#1}%
   \xdef\font@name{%
      \csname\curr@fontshape/\f@size\endcsname}%
   \pickup@font
   \font@name
   \@@enc@update}
```

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

For a detailed discussion see pr/3160.

```
\let\hmode@start@before@group\leavevmode
\def\DeclareTextSymbolDefault#1#2{%
   \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}
\def\DeclareTextAccentDefault#1#2{%
   \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}
\@onlypreamble\DeclareTextSymbolDefault
\@onlypreamble\DeclareTextAccentDefault
```

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

```
\def\UndeclareTextCommand#1#2{%
  \expandafter\ifx\csname#2\string#1\endcsname\relax
  \else
     \global\expandafter\let\csname#2\string#1\endcsname
            \@undefined
     \expandafter\expandafter\expandafter
     \ifx\expandafter\@thirdofthree#1\@undefined
       \expandafter\gdef\expandafter#1\expandafter
          {\csname ?-cmd\expandafter\endcsname\expandafter
           #1\csname?\string#1\endcsname}%
     \fi
 \fi
}
\@onlypreamble\UndeclareTextCommand
```

11.3.40 Hyphenation

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

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

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

11.3.5 Default encodings

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

The naming conventions in the kernel are not what we would use if we were starting from scratch... Those defined by DEK (like **\ae** and **\ss**) or by the TEX Users Group Technical Working Group on multi-lingual typesetting (like **\th** and **\ng**) have short names. Those which were added to the kernel in 1993 and early 1994 are named after their Adobe glyph names (like **\guillemotleft** and **\quotedblbase**). Unfortunately, this naming scheme won't work for all glyphs, since some names (like **\space**) are already used, and some (like **\endash**) are very likely to be defined by users. So we're now using the naming scheme of **\text** followed by the Adobe name, (like **\textendash** and **\textsterling**). Except that some glyphs don't have Adobe names, so we're using the names used by fontinst for those (like **\textcompwordmark**). Sigh. Some accents from OT1:

```
\DeclareTextAccentDefault{\"}{OT1}
\DeclareTextAccentDefault{\'}{OT1}
\DeclareTextAccentDefault{\.}{OT1}
\DeclareTextAccentDefault{\=}{OT1}
```

```
\DeclareTextAccentDefault{\H}{OT1}
\DeclareTextAccentDefault{\^}{OT1}
\DeclareTextAccentDefault{\`}{OT1}
\DeclareTextAccentDefault{\b}{OT1}
\DeclareTextAccentDefault{\c}{OT1}
\DeclareTextAccentDefault{\d}{OT1}
\DeclareTextAccentDefault{\r}{OT1}
\DeclareTextAccentDefault{\u}{OT1}
\DeclareTextAccentDefault{\v}{OT1}
\DeclareTextAccentDefault{\~}{OT1}
```

Some symbols from OT1:

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

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

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

```
\DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
\DeclareTextSymbolDefault{\textbackslash}{OMS}
\DeclareTextSymbolDefault{\textbar}{OMS}
\DeclareTextSymbolDefault{\textbardbl}{OMS}
\DeclareTextSymbolDefault{\textbraceleft}{OMS}
\DeclareTextSymbolDefault{\textbraceright}{OMS}
\DeclareTextSymbolDefault{\textbullet}{OMS}
\DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
\DeclareTextSymbolDefault{\textdagger}{OMS}
\DeclareTextSymbolDefault{\textparagraph}{OMS}
\DeclareTextSymbolDefault{\textperiodcentered}{OMS}
\DeclareTextSymbolDefault{\textsection}{OMS}
\DeclareTextAccentDefault{\textcircled}{OMS}
```

```
\DeclareTextSymbolDefault{\textless}{OML}
\DeclareTextSymbolDefault{\textgreater}{OML}
\DeclareTextAccentDefault{\t}{OML}
\DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
% \expandafter\def\expandafter
%                 ↊
      ↉\copyright\expandafter{\expandafter{\copyright}}
\DeclareTextCommandDefault{\textasciicircum}{\^{}}
\DeclareTextCommandDefault{\textasciitilde}{\~{}}
\DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
\DeclareTextCommandDefault{\textunderscore}{%
  \leavevmode \kern.06em\vbox{\hrule\@width.3em}}
\DeclareTextCommandDefault{\textvisiblespace}{%
   \mbox{\kern.06em\vrule \@height.3ex}%
   \vbox{\hrule \@width.3em}%
   \hbox{\vrule \@height.3ex}}
```

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

```
\DeclareTextCommandDefault{\textellipsis}{%
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font}
%\DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape ↊
      ↉r}}
\DeclareTextCommandDefault{\textregistered}{\textcircled{%
     \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont ↊
          ↉R}}
\DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
\DeclareTextCommandDefault{\SS}{SS}
\DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
\DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}
\DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
\DeclareRobustCommand{\{}{\ifmmode\lbrace\else\textbraceleft\fi}
\DeclareRobustCommand{\}}{\ifmmode\rbrace\else\textbraceright\fi}
\DeclareRobustCommand{\P}{\ifmmode\mathparagraph\else\textparagraph\fi}
\DeclareRobustCommand{\S}{\ifmmode\mathsection\else\textsection\fi}
\DeclareRobustCommand{\dag}{\ifmmode{\dagger}\else\textdagger\fi}
\DeclareRobustCommand{\ddag}{\ifmmode{\ddagger}\else\textdaggerdbl\fi}
\DeclareRobustCommand{\_}{%
   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
\DeclareRobustCommand{\copyright}{%
   \ifmmode{\nfss@text{\textcopyright}}\else\textcopyright\fi}
\DeclareRobustCommand{\pounds}{%
   \ifmmode\mathsterling\else\textsterling\fi}
\DeclareRobustCommand{\dots}{%
   \ifmmode\mathellipsis\else\textellipsis\fi}
\let\ldots\dots
%</2ekernel>
```

11.3.6 Definitions for the OT1 encoding

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

Declare the encoding.

```
%<*OT1>
\DeclareFontEncoding{OT1}{}{}
```

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

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

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

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

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

Some symbols which are faked from others:

```
% \DeclareTextCommand{\aa}{OT1}
%    {{\accent23a}}
\DeclareTextCommand{\L}{OT1}
   {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
\DeclareTextCommand{\l}{OT1}
   {\hmode@bgroup\@xxxii l\egroup}
% \DeclareTextCommand{\AA}{OT1}
%    {{\leavevmode\setbox\z@\hbox{h}\dimen@\ht\z@\advance\dimen@-1ex%
%     \rlap{\raise.67\dimen@\hbox{\char23}}A}
\DeclareTextCompositeCommand{\r}{OT1}{A}
   {\leavevmode\setbox\z@\hbox{!}\dimen@\ht\z@\advance\dimen@-1ex%
    \rlap{\raise.67\dimen@\hbox{\char23}}A}
\DeclareTextCommand{\ij}{OT1}{%
  \nobreak\hskip\z@skip i\kern-0.02em j\nobreak\hskip\z@skip}
\DeclareTextCommand{\IJ}{OT1}{%
  \nobreak\hskip\z@skip I\kern-0.02em J\nobreak\hskip\z@skip}
\DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \slshape
   \else
      \upshape
   \fi
   \char`\$\egroup}
\DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \itshape
   \else
      \fontshape{ui}\selectfont
   \fi
   \char`\$\egroup}
```

Here we are adding some more composite commands to the **OT1** encoding. This makes the use of certain accents with **i** compatible with their use with the **T1** encoding; this enables them to become true LaTeX internal representations. However, it will make these accents work a little less fast since a check will always be made for the existence of a composite.

```
\DeclareTextComposite{\.}{OT1}{i}{`\i}
\DeclareTextComposite{\.}{OT1}{\i}{`\i}
\DeclareTextCompositeCommand{\`}{OT1}{i}{\@tabacckludge`\i}
\DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
\DeclareTextCompositeCommand{\^}{OT1}{i}{\^\i}
\DeclareTextCompositeCommand{\"}{OT1}{i}{\"\i}
%</OT1>
```

11.3.7 Definitions for the T1 encoding

```
%<*T1>
\DeclareFontEncoding{T1}{}{}
```

```
\DeclareTextAccent{\`}{T1}{0}
\DeclareTextAccent{\'}{T1}{1}
\DeclareTextAccent{\^}{T1}{2}
\DeclareTextAccent{\~}{T1}{3}
\DeclareTextAccent{\"}{T1}{4}
\DeclareTextAccent{\H}{T1}{5}
\DeclareTextAccent{\r}{T1}{6}
\DeclareTextAccent{\v}{T1}{7}
\DeclareTextAccent{\u}{T1}{8}
\DeclareTextAccent{\=}{T1}{9}
\DeclareTextAccent{\.}{T1}{10}
```

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

Some symbols are constructed.

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

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

```
\DeclareTextSymbol{\IJ}{T1}{156}
\DeclareTextSymbol{\l}{T1}{170}
\DeclareTextSymbol{\ng}{T1}{173}
\DeclareTextSymbol{\oe}{T1}{247}
\DeclareTextSymbol{\o}{T1}{248}
\DeclareTextSymbol{\quotedblbase}{T1}{18}
\DeclareTextSymbol{\quotesinglbase}{T1}{13}
\DeclareTextSymbol{\ss}{T1}{255}
\DeclareTextSymbol{\textasciicircum}{T1}{`\^}
\DeclareTextSymbol{\textasciitilde}{T1}{`\~}
\DeclareTextSymbol{\textbackslash}{T1}{`\\}
\DeclareTextSymbol{\textbar}{T1}{`\|}
\DeclareTextSymbol{\textbraceleft}{T1}{`\{}
\DeclareTextSymbol{\textbraceright}{T1}{`\}}
\DeclareTextSymbol{\textcompwordmark}{T1}{23}
\DeclareTextSymbol{\textdollar}{T1}{`\$}
\DeclareTextSymbol{\textemdash}{T1}{22}
\DeclareTextSymbol{\textendash}{T1}{21}
\DeclareTextSymbol{\textexclamdown}{T1}{189}
\DeclareTextSymbol{\textgreater}{T1}{`\>}
%\DeclareTextSymbol{\texthyphenchar}{T1}{127}
%\DeclareTextSymbol{\texthyphen}{T1}{`\-}
\DeclareTextSymbol{\textless}{T1}{`\<}
\DeclareTextSymbol{\textquestiondown}{T1}{190}
\DeclareTextSymbol{\textquotedblleft}{T1}{16}
\DeclareTextSymbol{\textquotedblright}{T1}{17}
\DeclareTextSymbol{\textquotedbl}{T1}{`\"}
\DeclareTextSymbol{\textquoteleft}{T1}{`\`}
\DeclareTextSymbol{\textquoteright}{T1}{`\'}
\DeclareTextSymbol{\textsection}{T1}{159}
\DeclareTextSymbol{\textsterling}{T1}{191}
\DeclareTextSymbol{\textunderscore}{T1}{95}
\DeclareTextSymbol{\textvisiblespace}{T1}{32}
\DeclareTextSymbol{\th}{T1}{254}

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

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

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

\DeclareTextComposite{\v}{T1}{R}{144}
\DeclareTextComposite{\'}{T1}{S}{145}
```

```
\DeclareTextComposite{\v}{T1}{S}{146}
\DeclareTextComposite{\c}{T1}{S}{147}
\DeclareTextComposite{\v}{T1}{T}{148}
\DeclareTextComposite{\c}{T1}{T}{149}
\DeclareTextComposite{\H}{T1}{U}{150}
\DeclareTextComposite{\r}{T1}{U}{151}

\DeclareTextComposite{\"}{T1}{Y}{152}
\DeclareTextComposite{\'}{T1}{Z}{153}
\DeclareTextComposite{\v}{T1}{Z}{154}
\DeclareTextComposite{\.}{T1}{Z}{155}
\DeclareTextComposite{\.}{T1}{I}{157}

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

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

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

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

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

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

```
\DeclareTextComposite{\"}{T1}{I}{207}

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

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

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

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

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

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

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

%</T1>

%<*OMS>
\DeclareFontEncoding{OMS}{}{}

% \footnotechanges{v1.99}{2004/02/02}{Added \cs{textbigcircle}}
%    Note that slot 13 has in places been named |\Orb|: please root
```

```
%    out and destroy this impolity wherever you find it!
\begin{teX}
\DeclareTextSymbol{\textasteriskcentered}{OMS}{3}   % "03
\DeclareTextSymbol{\textbackslash}{OMS}{110}        % "6E
\DeclareTextSymbol{\textbar}{OMS}{106}              % "6A
\DeclareTextSymbol{\textbardbl}{OMS}{107}           % "6B
\DeclareTextSymbol{\textbraceleft}{OMS}{102}        % "66
\DeclareTextSymbol{\textbraceright}{OMS}{103}       % "67
\DeclareTextSymbol{\textbullet}{OMS}{15}            % "0F
\DeclareTextSymbol{\textdaggerdbl}{OMS}{122}        % "7A
\DeclareTextSymbol{\textdagger}{OMS}{121}           % "79
\DeclareTextSymbol{\textparagraph}{OMS}{123}        % "7B
\DeclareTextSymbol{\textperiodcentered}{OMS}{1}     % "01
\DeclareTextSymbol{\textsection}{OMS}{120}          % "78
\DeclareTextSymbol{\textbigcircle}{OMS}{13}         % "0D
\DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
   \ooalign{%
      \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
      \char 13 % "0D
   }%
 \egroup}
%</OMS>
```

11.3.8 Definitions for the OML encoding

[19]

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.OML

Declare the encoding.

```
%<*OML>
\DeclareFontEncoding{OML}{}{}
```

Declare the symbols.

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

11.3.9 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`;

[19] v1.6d 1994/10/30 Added the OML encoding.

Font files `ftp://ftp.gust.org.pl/TeX/language/polish/pl-tfm.zip`.

Declare the encoding.

```
%<*OT4>
\DeclareFontEncoding{OT4}{}{}
\DeclareFontSubstitution{OT4}{cmr}{m}{n}
```

Declare the accents.

```
\DeclareTextAccent{\"}{OT4}{127}
\DeclareTextAccent{\'}{OT4}{19}
\DeclareTextAccent{\.}{OT4}{95}
\DeclareTextAccent{\=}{OT4}{22}
\DeclareTextAccent{\^}{OT4}{94}
\DeclareTextAccent{\`}{OT4}{18}
\DeclareTextAccent{\~}{OT4}{126}
\DeclareTextAccent{\H}{OT4}{125}
\DeclareTextAccent{\u}{OT4}{21}
\DeclareTextAccent{\v}{OT4}{20}
\DeclareTextAccent{\r}{OT4}{23}
```

The ogonek accent is available only under a e A & E. But we have to provide some definition for `\k`. Some other accents have to be built by hand as in OT1: [20]

```
\DeclareTextCommand{\k}{OT4}[1]{%
    \TextSymbolUnavailable{\k{#1}}#1}
```

In these definitions we no longer use the helper function `\sh@ft` from plain.tex since that now has two incompatible definitions. [21]

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

Declare the text symbols.

```
\DeclareTextSymbol{\AE}{OT4}{29}
\DeclareTextSymbol{\OE}{OT4}{30}
\DeclareTextSymbol{\O}{OT4}{31}
\DeclareTextSymbol{\L}{OT4}{138}
\DeclareTextSymbol{\ae}{OT4}{26}
\DeclareTextSymbol{\guillemotleft}{OT4}{174}
\DeclareTextSymbol{\guillemotright}{OT4}{175}
\DeclareTextSymbol{\i}{OT4}{16}
\DeclareTextSymbol{\j}{OT4}{17}
\DeclareTextSymbol{\l}{OT4}{170}
\DeclareTextSymbol{\o}{OT4}{28}
\DeclareTextSymbol{\oe}{OT4}{27}
\DeclareTextSymbol{\quotedblbase}{OT4}{255}
```

[20] v1.9y 2000/01/30 Use `\hmode@bgroup` where applicable (pr/3160)
[21] v1.99g 2005/09/27 Replace `\sh@ft` by `\ltx@sh@ft`

```
\DeclareTextSymbol{\ss}{OT4}{25}
\DeclareTextSymbol{\textemdash}{OT4}{124}
\DeclareTextSymbol{\textendash}{OT4}{123}
\DeclareTextSymbol{\textexclamdown}{OT4}{60}
%\DeclareTextSymbol{\texthyphenchar}{OT4}{`\-}
%\DeclareTextSymbol{\texthyphen}{OT4}{`\-}
\DeclareTextSymbol{\textquestiondown}{OT4}{62}
\DeclareTextSymbol{\textquotedblleft}{OT4}{92}
\DeclareTextSymbol{\textquotedblright}{OT4}{`\"}
\DeclareTextSymbol{\textquoteleft}{OT4}{`\`}
\DeclareTextSymbol{\textquoteright}{OT4}{`\'}
```

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

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

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

11.3.10 Definitions for the TS1 encoding

```
%<*TS1>
\DeclareFontEncoding{TS1}{}{}
\DeclareFontSubstitution{TS1}{cmr}{m}{n}
```

```
\DeclareTextCommand{\capitalcedilla}{TS1}[1]
   {\hmode@bgroup
    \ooalign{\null#1\crcr\hidewidth\char11\hidewidth}\egroup}
\DeclareTextCommand{\capitalogonek}{TS1}[1]
   {\hmode@bgroup
    \ooalign{\null#1\crcr\hidewidth\char12\hidewidth}\egroup}
```

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

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

Tie accents.

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

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

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

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

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

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

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

```
\DeclareTextSymbol{\textasteriskcentered}{TS1}{42}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\DeclareTextSymbol{\textguarani}{TS1}{144}
\DeclareTextSymbol{\textpeso}{TS1}{145}
\DeclareTextSymbol{\textlira}{TS1}{146}
\DeclareTextSymbol{\textrecipe}{TS1}{147}
```

```
\DeclareTextSymbol{\textinterrobang}{TS1}{148}
\DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
\DeclareTextSymbol{\textdong}{TS1}{150}
\DeclareTextSymbol{\texttrademark}{TS1}{151}

\DeclareTextSymbol{\textpertenthousand}{TS1}{152}
\DeclareTextSymbol{\textpilcrow}{TS1}{153}
\DeclareTextSymbol{\textbaht}{TS1}{154}
\DeclareTextSymbol{\textnumero}{TS1}{155}

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

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

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

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

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

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

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

\DeclareTextSymbol{\textdiv}{TS1}{246}
%</TS1>

%<*package>
```

```
\def\update@uclc@with@cyrillic{%
 \expandafter\def\expandafter\@uclclist\expandafter
  {\@uclclist
  \cyra\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
  \CYRABHDZE\cyrabhha\CYRABHHA\cyrae\CYRAE\cyrb\CYRB\cyrbyus
  \CYRBYUS\cyrc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrdsc
  \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
  \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
  \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
  \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
  \cyrghcrs\CYRGHCRS\cyrghk\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
  \cyrhdsc\CYRHDSC\cyrhhcrs\CYRHHCRS\cyrhhk\CYRHHK\cyrhrdsn
  \CYRHRDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
  \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
  \cyrkbeak\CYRKBEAK\cyrkdsc\CYRKDSC\cyrkhcrs\CYRKHCRS\cyrkhk
  \CYRKHK\cyrkvcrs\CYRKVCRS\cyrl\CYRL\cyrldsc\CYRLDSC\cyrlhk
  \CYRLHK\cyrlje\CYRLJE\cyrm\CYRM\cyrmdsc\CYRMDSC\cyrmhk\CYRMHK
  \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnhk\CYRNHK\cyrnje
  \CYRNJE\cyrnlhk\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
  \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
  \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
  \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
  \cyrsftsn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
  \cyrt\CYRT\cyrtdsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
  \cyru\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyry\CYRY
  \cyrya\CYRYA\cyryat\CYRYAT\cyryhcrs\CYRYHCRS\cyryi\CYRYI\cyryo
  \CYRYO\cyryu\CYRYU\cyrz\CYRZ\cyrzdsc\CYRZDSC\cyrzh\CYRZH
  \cyrzhdsc\CYRZHDSC}%
 \let\update@uclc@with@cyrillic\relax
}
\DeclareOption*{%
   \let\encodingdefault\CurrentOption
   \edef\reserved@f{%
     \lowercase{\def\noexpand\reserved@f{\CurrentOption
          enc.def}}}%
   \reserved@f
   \InputIfFileExists\reserved@f
        {}{\PackageError{fontenc}%
         {Encoding file `\reserved@f' not found.%
          \MessageBreak
           You might have misspelt the name of the encoding}%
         {Necessary code for this encoding was not
          loaded.\MessageBreak
          Thus calling the encoding later on will
          produce further error messages.}}%
  \let\reserved@f\relax
  \expandafter\in@\expandafter{\CurrentOption}%
                              {T2A,T2B,T2C,X2,LCY,OT2}%
  \ifin@
     \expandafter\in@\expandafter\cyra\expandafter
                              {\@uclclist}%
     \ifin@
     \else
```

```
        \update@uclc@with@cyrillic
      \fi
  \fi
}
\ProcessOptions*
\fontencoding\encodingdefault\selectfont
\let\update@uclc@with@cyrillic\relax
\global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
\global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
\global\let\@ifl@ter@@\@ifl@ter
\def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
%</package>
```

11.3.11 The textcomp package

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

```
%<*TS1sty>
```

Since many PostScript fonts only implement a subset of **TS1** many commands only produce black blobs of ink. To resolve the resulting problems a number of options have been introduced and some code has been developed to distinguish sub-encodings.

The sub-encodings have a numerical id and are defined as follows for TS1:

#5 those TS1 symbols that are also in the ISO-Adobe character set; without **textcurrency**, which is often misused for the Euro. Older Type1 fonts from the non-TEX world provide only this subset.

#4 = #5 + **\texteuro**. Most newer fonts provide this.

#3 = #4 + **\textomega**. Can also be described as TS1∩(ISO-Adobe∪MacRoman). (Except for the missing "currency".)

#2 = #3 + **\textestimated** + **\textcurrency**. Can also be described as TS1 ∩ Adobe-Western-2. This may be relevant for OpenType fonts, which usually show the Adobe-Western-2 character set.

#1 = TS1 without **\textcircled** and **\t**. These two glyphs are often not implemented and if their kernel defaults are changed commands like **\copyright** unnecessarily fail.

#0 = full TS1

And here a summary to go in the transcript file:

```
\PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
    \space\space 5 = only ISO-Adobe without
                                \string\textcurrency\MessageBreak
```

```
    \space\space 4 = 5 + \string\texteuro\MessageBreak
    \space\space 3 = 4 + \string\textohm\MessageBreak
    \space\space 2 = 3 + \noexpand\textestimated+
                                \string\textcurrency\MessageBreak
    \space\space 1 = TS1 - \noexpand\textcircled-
                                            \string\t\MessageBreak
    \space\space 0 = TS1 (full)\MessageBreak
    Font families with sub-encoding setting ↩
         ↪implement\MessageBreak
    only a restricted character set as indicated.\MessageBreak
    Family '?' is the default used for unknown ↩
         ↪fonts.\MessageBreak
    See the documentation for details\@gobble}
```

`\DeclareEncodingSubset` {⟨*encoding*⟩}{⟨*family name*⟩}{⟨*subset encoding*⟩}

An encoding subset to which a font family belongs is declared by the command `\DeclareEncodingSubset` that takes the major encoding as the first argument (e.g., `TS1`), the family name as the second argument (e.g., `cmr`), and the subset encoding id as a third, (e.g., `0` for `cmr`).
The default encoding subset to use when nothing is known about the current font family is named `?`.

```
\def\DeclareEncodingSubset#1#2#3{%
   \@ifundefined{#1:#2}%
     {\PackageInfo{textcomp}{Setting #2 sub-encoding to ↩
          ↪#1/#3}}%
     {\PackageInfo{textcomp}{Changing #2 sub-encoding to ↩
          ↪#1/#3}}%
   \@namedef{#1:#2}{#3}}
\@onlypreamble\DeclareEncodingSubset
```

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).

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

This is implemented by defining the default subset:

```
\DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
\DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
\DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
\DeclareOption{safe}{\DeclareEncodingSubset{TS1}{?}{5}}
```

The default is "almostfull" which means that old documents will work except that `\textcircled` and `\t` will use the kernel defaults (with the advantage that this also works if the current font (as often the case) doesn't implement these glyphs.

The "force" option simply sets the switch to true.

```
\DeclareOption{force}{\tc@forcedtrue}
```

The suggestions to user is to use the "safe" option always unless that balks in which case they could switch to "almostfull" but then better check their output manually.

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

```
\ExecuteOptions{almostfull}
\ProcessOptions\relax
```

`\CheckEncodingSubset`

The command `\CheckEncodingSubset` will check if the current font family has the right encoding subset to typeset a certain command. It takes five arguments as follows: first argument is either `\UseTextSymbol`, `\UseTextAccent` depending on whether or not the symbol is a text symbol or a text accent.
The second argument is the encoding from which this symbol should be fetched.
The third argument is either a fake accessor command or an error message. the code in that argument (if ever executed) receives two arguments: `#2` and `#5` of `\CheckEncodingSubset`.
Argument four is the subset encoding id to test against: if this value is higher than the subset id of the current font family then we typeset the symbol, i.e., execute `#1{#2}#5` otherwise it runs `#3#5`, e.g., to produce an error message or fake the glyph somehow.
Argument five is the symbol or accent command that is being checked.
For usage examples see definitions below.

```
\iftc@forced
```

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

```
\else
\def\CheckEncodingSubset#1#2#3#4#5{%
    \ifnum #4>%
```

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

\def\tc@subst#1{%
   \tc@errorwarn{textcomp}%  % should be latex error if general
    {Symbol \string#1 not provided by\MessageBreak
     font family \f@family\space
     in TS1 encoding.\MessageBreak Default family used instead}\@eha
  \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
}
\def\textcompsubstdefault{cmr}
% error commands take argument:
% #1 symbol to be used
\def\tc@error#1{%
   \PackageError{textcomp}%  % should be latex error if general
    {Accent \string#1 not provided by\MessageBreak
     font family \f@family\space
     in TS1 encoding}\@eha
}
\def\tc@fake@euro#1{%
   \leavevmode
   \PackageInfo{textcomp}{Faking \noexpand#1for font family
                          \f@family\MessageBreak in TS1 encoding}%
   \valign{##\cr
      \vfil\hbox to 0.07em{\dimen@\f@size\p@
                           \math@fontsfalse
                           \fontsize{.7\dimen@}\z@\selectfont=\hss}%
      \vfil\cr%
      \hbox{C}\crcr
   }%
}
```

These are two abbreviations that we use below to check symbols and accents in TS1. Only there to save some space, e.g., we can then write

```
\DeclareTextCommandDefault{\textcurrency}{\tc@check@symbol3\textcurrency}
```

to ensure that `\textcurrency` is only typeset if the current font has a TS1 subset id of less than 3. Otherwise `\tc@error` is called telling the user that for this font family `\textcurreny` is not available.

```
\def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
\def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
```

We start with the commands that are "safe" and which can be unconditionally set up, first the accents...

```
\DeclareTextAccentDefault{\capitalcedilla}{TS1}
\DeclareTextAccentDefault{\capitalogonek}{TS1}
\DeclareTextAccentDefault{\capitalgrave}{TS1}
\DeclareTextAccentDefault{\capitalacute}{TS1}
\DeclareTextAccentDefault{\capitalcircumflex}{TS1}
\DeclareTextAccentDefault{\capitaltilde}{TS1}
\DeclareTextAccentDefault{\capitaldieresis}{TS1}
\DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
\DeclareTextAccentDefault{\capitalring}{TS1}
\DeclareTextAccentDefault{\capitalcaron}{TS1}
\DeclareTextAccentDefault{\capitalbreve}{TS1}
\DeclareTextAccentDefault{\capitalmacron}{TS1}
\DeclareTextAccentDefault{\capitaldotaccent}{TS1}
```

Á

```
\DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
\DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
\DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
\DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
\DeclareTextSymbolDefault{\texttwelveudash}{TS1}
\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
\DeclareTextSymbolDefault{\textdollar}{TS1}
\DeclareTextSymbolDefault{\textquotesingle}{TS1}
\DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
\DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
\DeclareTextSymbolDefault{\textminus}{TS1}
\DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
\DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
\DeclareTextSymbolDefault{\textasciigrave}{TS1}
\DeclareTextSymbolDefault{\texttildelow}{TS1}
\DeclareTextSymbolDefault{\textasciibreve}{TS1}
\DeclareTextSymbolDefault{\textasciicaron}{TS1}
\DeclareTextSymbolDefault{\textgravedbl}{TS1}
\DeclareTextSymbolDefault{\textacutedbl}{TS1}
\DeclareTextSymbolDefault{\textdagger}{TS1}
\DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
\DeclareTextSymbolDefault{\textbardbl}{TS1}
\DeclareTextSymbolDefault{\textperthousand}{TS1}
\DeclareTextSymbolDefault{\textbullet}{TS1}
\DeclareTextSymbolDefault{\textcelsius}{TS1}
\DeclareTextSymbolDefault{\textflorin}{TS1}
\DeclareTextSymbolDefault{\texttrademark}{TS1}
\DeclareTextSymbolDefault{\textcent}{TS1}
\DeclareTextSymbolDefault{\textsterling}{TS1}
\DeclareTextSymbolDefault{\textyen}{TS1}
\DeclareTextSymbolDefault{\textbrokenbar}{TS1}
\DeclareTextSymbolDefault{\textsection}{TS1}
\DeclareTextSymbolDefault{\textasciidieresis}{TS1}
\DeclareTextSymbolDefault{\textcopyright}{TS1}
```

```
\DeclareTextSymbolDefault{\textordfeminine}{TS1}
\DeclareTextSymbolDefault{\textlnot}{TS1}
\DeclareTextSymbolDefault{\textregistered}{TS1}
\DeclareTextSymbolDefault{\textasciimacron}{TS1}
\DeclareTextSymbolDefault{\textdegree}{TS1}
\DeclareTextSymbolDefault{\textpm}{TS1}
\DeclareTextSymbolDefault{\texttwosuperior}{TS1}
\DeclareTextSymbolDefault{\textthreesuperior}{TS1}
\DeclareTextSymbolDefault{\textasciiacute}{TS1}
\DeclareTextSymbolDefault{\textmu}{TS1}
\DeclareTextSymbolDefault{\textparagraph}{TS1}
\DeclareTextSymbolDefault{\textperiodcentered}{TS1}
\DeclareTextSymbolDefault{\textonesuperior}{TS1}
\DeclareTextSymbolDefault{\textordmasculine}{TS1}
\DeclareTextSymbolDefault{\textonequarter}{TS1}
\DeclareTextSymbolDefault{\textonehalf}{TS1}
\DeclareTextSymbolDefault{\textthreequarters}{TS1}
\DeclareTextSymbolDefault{\texttimes}{TS1}
\DeclareTextSymbolDefault{\textdiv}{TS1}

\DeclareTextCommandDefault{\texteuro}
   {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

\DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}

\DeclareTextCommandDefault{\textestimated}%
    {\tc@check@symbol3\textestimated}
\DeclareTextCommandDefault{\textcurrency}%
    {\tc@check@symbol3\textcurrency}

\DeclareTextCommandDefault{\capitaltie}%
    {\tc@check@accent2\capitaltie}
\DeclareTextCommandDefault{\newtie}%
    {\tc@check@accent2\newtie}
\DeclareTextCommandDefault{\capitalnewtie}%
    {\tc@check@accent2\capitalnewtie}
\DeclareTextCommandDefault{\textleftarrow}%
    {\tc@check@symbol2\textleftarrow}
\DeclareTextCommandDefault{\textrightarrow}%
    {\tc@check@symbol2\textrightarrow}
\DeclareTextCommandDefault{\textblank}%
    {\tc@check@symbol2\textblank}
\DeclareTextCommandDefault{\textdblhyphen}%
    {\tc@check@symbol2\textdblhyphen}
\DeclareTextCommandDefault{\textzerooldstyle}%
    {\tc@check@symbol2\textzerooldstyle}
\DeclareTextCommandDefault{\textoneoldstyle}%
    {\tc@check@symbol2\textoneoldstyle}
\DeclareTextCommandDefault{\texttwooldstyle}%
    {\tc@check@symbol2\texttwooldstyle}
\DeclareTextCommandDefault{\textthreeoldstyle}%
    {\tc@check@symbol2\textthreeoldstyle}
\DeclareTextCommandDefault{\textfouroldstyle}%
    {\tc@check@symbol2\textfouroldstyle}
\DeclareTextCommandDefault{\textfiveoldstyle}%
    {\tc@check@symbol2\textfiveoldstyle}
```

```
\DeclareTextCommandDefault{\textsixoldstyle}%
    {\tc@check@symbol2\textsixoldstyle}
\DeclareTextCommandDefault{\textsevenoldstyle}%
    {\tc@check@symbol2\textsevenoldstyle}
\DeclareTextCommandDefault{\texteightoldstyle}%
    {\tc@check@symbol2\texteightoldstyle}
\DeclareTextCommandDefault{\textnineoldstyle}%
    {\tc@check@symbol2\textnineoldstyle}
\DeclareTextCommandDefault{\textlangle}%
    {\tc@check@symbol2\textlangle}
\DeclareTextCommandDefault{\textrangle}%
    {\tc@check@symbol2\textrangle}
\DeclareTextCommandDefault{\textmho}%
    {\tc@check@symbol2\textmho}
\DeclareTextCommandDefault{\textbigcircle}%
    {\tc@check@symbol2\textbigcircle}
\DeclareTextCommandDefault{\textuparrow}%
    {\tc@check@symbol2\textuparrow}
\DeclareTextCommandDefault{\textdownarrow}%
    {\tc@check@symbol2\textdownarrow}
\DeclareTextCommandDefault{\textborn}%
    {\tc@check@symbol2\textborn}
\DeclareTextCommandDefault{\textdivorced}%
    {\tc@check@symbol2\textdivorced}
\DeclareTextCommandDefault{\textdied}%
    {\tc@check@symbol2\textdied}
\DeclareTextCommandDefault{\textleaf}%
    {\tc@check@symbol2\textleaf}
\DeclareTextCommandDefault{\textmarried}%
    {\tc@check@symbol2\textmarried}
\DeclareTextCommandDefault{\textmusicalnote}%
    {\tc@check@symbol2\textmusicalnote}
\DeclareTextCommandDefault{\textdblhyphenchar}%
    {\tc@check@symbol2\textdblhyphenchar}
\DeclareTextCommandDefault{\textdollaroldstyle}%
    {\tc@check@symbol2\textdollaroldstyle}
\DeclareTextCommandDefault{\textcentoldstyle}%
    {\tc@check@symbol2\textcentoldstyle}
\DeclareTextCommandDefault{\textcolonmonetary}%
    {\tc@check@symbol2\textcolonmonetary}
\DeclareTextCommandDefault{\textwon}%
    {\tc@check@symbol2\textwon}
\DeclareTextCommandDefault{\textnaira}%
    {\tc@check@symbol2\textnaira}
\DeclareTextCommandDefault{\textguarani}%
    {\tc@check@symbol2\textguarani}
\DeclareTextCommandDefault{\textpeso}%
    {\tc@check@symbol2\textpeso}
\DeclareTextCommandDefault{\textlira}%
    {\tc@check@symbol2\textlira}
\DeclareTextCommandDefault{\textrecipe}%
    {\tc@check@symbol2\textrecipe}
\DeclareTextCommandDefault{\textinterrobang}%
    {\tc@check@symbol2\textinterrobang}
```

```
\DeclareTextCommandDefault{\textinterrobangdown}%
    {\tc@check@symbol2\textinterrobangdown}
\DeclareTextCommandDefault{\textdong}%
    {\tc@check@symbol2\textdong}
\DeclareTextCommandDefault{\textpertenthousand}%
    {\tc@check@symbol2\textpertenthousand}
\DeclareTextCommandDefault{\textpilcrow}%
    {\tc@check@symbol2\textpilcrow}
\DeclareTextCommandDefault{\textbaht}%
    {\tc@check@symbol2\textbaht}
\DeclareTextCommandDefault{\textnumero}%
    {\tc@check@symbol2\textnumero}
\DeclareTextCommandDefault{\textdiscount}%
    {\tc@check@symbol2\textdiscount}
\DeclareTextCommandDefault{\textopenbullet}%
    {\tc@check@symbol2\textopenbullet}
\DeclareTextCommandDefault{\textservicemark}%
    {\tc@check@symbol2\textservicemark}
\DeclareTextCommandDefault{\textlquill}%
    {\tc@check@symbol2\textlquill}
\DeclareTextCommandDefault{\textrquill}%
    {\tc@check@symbol2\textrquill}
\DeclareTextCommandDefault{\textcopyleft}%
    {\tc@check@symbol2\textcopyleft}
\DeclareTextCommandDefault{\textcircledP}%
    {\tc@check@symbol2\textcircledP}
\DeclareTextCommandDefault{\textreferencemark}%
    {\tc@check@symbol2\textreferencemark}
\DeclareTextCommandDefault{\textsurd}%
    {\tc@check@symbol2\textsurd}
```

```
\DeclareTextCommandDefault{\textcircled}
  {\CheckEncodingSubset\UseTextAccent{TS1}%
    {\UseTextAccent{OMS}}1\textcircled}
\DeclareTextCommandDefault{\t}
  {\CheckEncodingSubset\UseTextAccent{TS1}%
    {\UseTextAccent{OML}}1\t}
```

```
\input{ts1enc.def}
```

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

```
%\UndeclareTextCommand{\textsterling}{OT4}
%\UndeclareTextCommand{\textdollar}  {OT4}
```

```
\UndeclareTextCommand{\textperthousand}{T1}
%\UndeclareTextCommand{\textpertenthousand}{T1}
```

```
\DeclareRobustCommand\oldstylenums[1]{%
 \begingroup
  \ifmmode
   \mathgroup\symletters #1%
  \else
   \CheckEncodingSubset\@use@text@encoding{TS1}%
       {\PackageWarning{textcomp}%
          {Oldstyle digits unavailable for
```

```
            family \f@family.\MessageBreak
            Lining digits used instead}}%
        \tw@{#1}%
    \fi
  \endgroup
}
```

11.3.110 Subset encoding defaults

For many font families commonly used in the TEX world we provide the subset encoding data here. Users can add additional font families in the file `textcomp.cfg` if they own other fonts.

However, if the option "forced" was given then all subset encoding specifications are ignored, so there is no point in setting any of them up:

```
\iftc@forced \else
```

```
\DeclareEncodingSubset{TS1}{cmr}      {0}
\DeclareEncodingSubset{TS1}{cmss}     {0}
\DeclareEncodingSubset{TS1}{cmtt}     {0}
\DeclareEncodingSubset{TS1}{cmvtt}    {0}
\DeclareEncodingSubset{TS1}{cmbr}     {0}
\DeclareEncodingSubset{TS1}{cmtl}     {0}
\DeclareEncodingSubset{TS1}{ccr}      {0}
```

```
\DeclareEncodingSubset{TS1}{ptm}      {4}
\DeclareEncodingSubset{TS1}{pcr}      {4}
\DeclareEncodingSubset{TS1}{phv}      {4}
\DeclareEncodingSubset{TS1}{ppl}      {3}
\DeclareEncodingSubset{TS1}{pag}      {4}
\DeclareEncodingSubset{TS1}{pbk}      {4}
\DeclareEncodingSubset{TS1}{pnc}      {4}
\DeclareEncodingSubset{TS1}{pzc}      {4}
\DeclareEncodingSubset{TS1}{bch}      {4}
\DeclareEncodingSubset{TS1}{put}      {5}
```

Other CTAN fonts (probably not complete):

```
\DeclareEncodingSubset{TS1}{uag}      {5}
\DeclareEncodingSubset{TS1}{ugq}      {5}
\DeclareEncodingSubset{TS1}{ul8}      {4}
\DeclareEncodingSubset{TS1}{ul9}      {4}  % (LuxiSans, one day)
\DeclareEncodingSubset{TS1}{augie}    {5}
\DeclareEncodingSubset{TS1}{dayrom}   {3}
\DeclareEncodingSubset{TS1}{dayroms}  {3}
\DeclareEncodingSubset{TS1}{pxr}      {0}
\DeclareEncodingSubset{TS1}{pxss}     {0}
\DeclareEncodingSubset{TS1}{pxtt}     {0}
\DeclareEncodingSubset{TS1}{txr}      {0}
\DeclareEncodingSubset{TS1}{txss}     {0}
\DeclareEncodingSubset{TS1}{txtt}     {0}
```

Latin Modern and TeX Gyre: [22] [23] [24]

```
\DeclareEncodingSubset{TS1}{lmr}      {0}
\DeclareEncodingSubset{TS1}{lmdh}     {0}
\DeclareEncodingSubset{TS1}{lmss}     {0}
\DeclareEncodingSubset{TS1}{lmssq}    {0}
\DeclareEncodingSubset{TS1}{lmvtt}    {0}
\DeclareEncodingSubset{TS1}{lmtt}     {0}
```

```
\DeclareEncodingSubset{TS1}{qhv}      {0}
\DeclareEncodingSubset{TS1}{qag}      {0}
\DeclareEncodingSubset{TS1}{qbk}      {0}
\DeclareEncodingSubset{TS1}{qcr}      {0}
\DeclareEncodingSubset{TS1}{qcs}      {0}
\DeclareEncodingSubset{TS1}{qpl}      {0}
\DeclareEncodingSubset{TS1}{qtm}      {0}
\DeclareEncodingSubset{TS1}{qzc}      {0}
\DeclareEncodingSubset{TS1}{qhvc}     {0}
```

Fourier-GUTenberg:

```
\DeclareEncodingSubset{TS1}{futs}     {4}
\DeclareEncodingSubset{TS1}{futx}     {4}
\DeclareEncodingSubset{TS1}{futj}     {4}
```

Y&Y's Lucida Bright

```
\DeclareEncodingSubset{TS1}{hlh}      {3}
\DeclareEncodingSubset{TS1}{hls}      {3}
\DeclareEncodingSubset{TS1}{hlst}     {3}
```

The remaining settings for Lucida are conservative: the following fonts contain the **\textohm** character but not the **\texteuro**, i.e., belong to neither subset 4 nor subset 3. If you want to use the **\textohm** with these fonts copy these definition to `textcomp.cfg` and change the subset to 3. However in that case make sure that you do not use the **\texteuro**.

```
\DeclareEncodingSubset{TS1}{hlct}     {5}
\DeclareEncodingSubset{TS1}{hlx}      {5}
\DeclareEncodingSubset{TS1}{hlce}     {5}
\DeclareEncodingSubset{TS1}{hlcn}     {5}
\DeclareEncodingSubset{TS1}{hlcw}     {5}
\DeclareEncodingSubset{TS1}{hlcf}     {5}
```

Other commercial families...

```
\DeclareEncodingSubset{TS1}{pplx}     {3}
\DeclareEncodingSubset{TS1}{pplj}     {3}
\DeclareEncodingSubset{TS1}{ptmx}     {4}
\DeclareEncodingSubset{TS1}{ptmj}     {4}
```

If the file `textcomp.cfg` exists it will be loaded at this point. This allows to define further subset encodings for font families not covered by default.

```
\InputIfFileExists{textcomp.cfg}
  {\PackageInfo{textcomp}{Local configuration file used}}{}
```

[22] v1.99k 2009/10/28 Added Latin Modern and TeX Gyre subsets
[23] v1.99l 2009/11/04 Added more Latin Modern and TeX Gyre subsets
[24] v1.99m 2015/02/16 Added lmtt (Heiko Oberdiek) latex/4415

```
\fi
%</TS1sty>
```

h a n g

12

Counters and Lengths

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed. These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002

16

-2.5

-2.1

0.87865

12.1 Counters and Lengths

Commands for defining and using counters.

??[→ P. ??] To define a new counter.

??[→ P. ??] To set the value of counters.

??[→ P. ??] Increase the counter `#1` by the number `#2`.

```
Example 1.1 test

1 \the\value{chapter}

12
```

```
2 %<*2ekernel>
```

12.1.1 Environment Counter Macros

An environment foo has an associated counter defined by the following control sequences:

`\c@foo` Contains the counter's numerical value. It is defined by `\newcount\foocounter`.

`\thefoo` Macro that expands to the printed value of `\foocounter`. For example, if sections are numbered within chapters, and section headings look like
Section II-3. The Nature of Counters
then `\thesection` might be defined by:
```
\def\thesection
    {\@Roman{\c@chapter}-
\@arabic{\c@section}}
```

`\p@foo` Macro that expands to a printed 'reference prefix' of counter foo. Any `\ref` to a value created by counter foo will produce the expansion of `\p@foo\thefoo` when the `\label` command is executed. See file `ltxref.dtx` for an extension of this mechanism.

`\cl@foo` List of counters to be reset when foo stepped. Has format `\@elt{countera}\@elt{counterb}\@elt{counterc}`.

NOTE:
`\thefoo` and `\p@foo` *must* be defined in such a way that `\edef\bar{\thefoo}` or `\edef\bar{\p@foo}` defines `\bar` so that it will evaluate to the counter value at the time of the `\edef`, even after `\foocounter` and any other counters have been changed. This will happen if you use the standard commands `\@arabic`, `\@Roman`, etc.

`\setcounter`

```
\def\setcounter#1#2{%
  \@ifundefined{c@#1}%
    {\@nocounterr{#1}}%
    {\global\csname c@#1\endcsname#2\relax}}
```

\addtocounter

Globally increments **\foocounter** by ⟨*val*⟩.

```
\def\addtocounter#1#2{%
  \@ifundefined{c@#1}%
    {\@nocounterr{#1}}%
    {\global\advance\csname c@#1\endcsname #2\relax}}
```

\newcounter

\newcounter{⟨*newctr*⟩}[⟨*oldctr*⟩] Defines ⟨*newctr*⟩ to be a counter, which is reset when counter ⟨*oldctr*⟩ is stepped. If ⟨*newctr*⟩ already defined

```
\def\newcounter#1{%
  \expandafter\@ifdefinable \csname c@#1\endcsname
    {\@definecounter{#1}}%
  \@ifnextchar[{\@newctr{#1}}{}}
```

\value

\value{⟨*ctr*⟩} produces the value of counter ⟨*ctr*⟩, for use with a **\setcounter** or **\addtocounter** command.

```
\def\value#1{\csname c@#1\endcsname}
```

\@newctr

```
\def\@newctr#1[#2]{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}
```

(*End definition for* `\@newctr`.)

\stepcounter

\stepcounterfoo Globally increments counter **\c@FOO** and resets all subsidiary counters.

```
\def\stepcounter#1{%
  \addtocounter{#1}\@ne
  \begingroup
    \let\@elt\@stpelt
    \csname cl@#1\endcsname
  \endgroup}
```

\@stpelt

Rather than resetting the "within" counter to zero we set it to −1 and then run **\stepcounter** that moves it to 0 and also initiates resetting the next level down.

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\@stpelt}
%<latexrelease>                                {Reset nested counters}%
%<*2ekernel|latexrelease>
\def\@stpelt#1{\global\csname c@#1\endcsname \m@ne\stepcounter{#1}}%
%<latexrelease>\EndIncludeInRelease
%</2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{0000/00/00}{\@stpelt}
%<latexrelease>                                {Reset nested counters}%%
%<latexrelease>\def\@stpelt#1{\global\csname c@#1\endcsname \z@}%
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

\cl@@ckpt

```
\def\cl@@ckpt{\@elt{page}}
```

\@definecounter⟨*counter name*⟩

The command defines a new counter. It uses the TeX's \newcount to allocate a counter starting from c@. It then sets the counter to zero.

\@addtoreset{⟨*counter*⟩} {⟨*counter*⟩}

```
\def\@definecounter#1{\expandafter\newcount\csname c@#1\endcsname
    \setcounter{#1}\z@
    \global\expandafter\let\csname cl@#1\endcsname\@empty
    \@addtoreset{#1}{@ckpt}%
    \global\expandafter\let\csname p@#1\endcsname\@empty
    \expandafter
    \gdef\csname the#1\expandafter\endcsname\expandafter
        {\expandafter\@arabic\csname c@#1\endcsname}}
\def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}}
```

Numbering commands for definitions of \theCOUNTER and \list arguments.

All commands can now be used in text and math mode.

\arabic

```
\def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}
```

\roman

```
\def\roman#1{\expandafter\@roman\csname c@#1\endcsname}
```

\Roman

```
\def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}
```

\alph

Representation of ⟨*counter*⟩ as a lower-case letter: 1 = a, 2 = b, etc.

```
\def\alph#1{\expandafter\@alph\csname c@#1\endcsname}
```

\Alph

```
\def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}
```

\fnsymbol

Representation of ⟨*COUNTER*⟩ as a footnote symbol: 1 = ∗, 2 = †, etc.

```
\def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}
```

\@arabic

```
\def\@arabic#1{\number #1}  %% changed 29 Apr 86
```

\@roman

```
\def\@roman#1{\romannumeral #1}
```

\@Roman

`\@Roman\FOOcounter` Representation of `\FOOcounter` as upper-case Roman numerals.

```
\def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}
```

\@slowromancap

Fully expandable macro to change a roman number to uppercase.

```
\def\@slowromancap#1{\ifx @#1% then terminate
     \else
       \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
       c#1C\else\if d#1D\else \if ↊
            ↊m#1M\else#1\fi\fi\fi\fi\fi\fi\fi
       \expandafter\@slowromancap
     \fi
}
```

\@alph

```
\def\@alph#1{%
  \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or ↊
        ↊j\or
   k\or l\or m\or n\or o\or p\or q\or r\or s\or t\or u\or ↊
        ↊v\or w\or x\or
    y\or z\else\@ctrerr\fi}
```

\@Alph

\@Alph\FOOcounter Representation of \FOOcounter as an upper-case letter: 1 = A, 2 = B, etc.

```
\def\@Alph#1{%
  \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
   K\or L\or M\or N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or X\or
    Y\or Z\else\@ctrerr\fi}
```

See also the dtx floats in LaTeX for further treatment of footnotes.

\@fnsymbol

This macro is another example of an ever recurring problem in TeX: Determining if something is text-mode or math-mode. It is imperative for the decision between text and math to be delayed until the actual typesetting is done as the code in question may go through an \edef or \write where an \ifmmode test would be executed prematurely. Hence in the implementation below, \@fnsymbol is not robust in itself but the parts doing the actual typesetting are.

In the case of \@fnsymbol we make use of the robust command \TextOrMath which takes two arguments and typesets the first if in text-mode and the second if in math-mode. Note that in order for this command to make the correct decision, it must insert a \relax token if run under regular TeX, which ruins any kerning between the preceding characters and whatever awaits typesetting. If you use eTeX as engine for LaTeX (as recommended) this unfortunate side effect is not present.

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TexOrMath}%
%<*2ekernel|latexrelease>
\def\@fnsymbol#1{%
   \ifcase#1\or \TextOrMath\textasteriskcentered *\or
   \TextOrMath \textdagger \dagger\or
   \TextOrMath \textdaggerdbl \ddagger \or
   \TextOrMath \textsection  \mathsection\or
   \TextOrMath \textparagraph \mathparagraph\or
   \TextOrMath \textbardbl \|\or
   \TextOrMath \{\textasteriskcentered\textasteriskcentered}{**}\or
   \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
   \TextOrMath \{\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
   \@ctrerr \fi
}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TexOrMath}%
%<latexrelease>\def\@fnsymbol#1{\ensuremath{%
```

```
%<latexrelease>   \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
%<latexrelease>     \mathparagraph\or \|\or **\or \dagger\dagger
%<latexrelease>     \or \ddagger\ddagger \else\@ctrerr\fi}}%
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

\TextOrMath{⟨*arg1*⟩}{⟨*arg2*⟩}

When using regular TEX, we make this command robust so that it always selects the correct branch in an **\ifmmode** switch with the usual disadvantage of ruining kerning. For the application we use it for here that shouldn't matter. The alternative would be to mimic **\IeC** from inputenc but then it wil have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of eTEX will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running eTEX but making sure not to permanently turn **\protected** into **\relax**.

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
%<*2ekernel|latexrelease>
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
```

In case of ordinary TEX we define **\TextOrMath** as a robust command but make sure it always grabs its arguments. If we didn't do this it might very well gobble spaces in the input stream.

```
\DeclareRobustCommand\TextOrMath{%
  \ifmmode  \expandafter\@secondoftwo
  \else     \expandafter\@firstoftwo  \fi}
\protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
\else
```

For eTEX the situation is similar. The robust macro is a hidden one so that we again avoid problems of gobbling spaces in the input.

```
\protected\expandafter\def\csname TextOrMath\space\endcsname{%
  \ifmmode  \expandafter\@secondoftwo
  \else     \expandafter\@firstoftwo  \fi}
\edef\TextOrMath#1#2{%
  \expandafter\noexpand\csname TextOrMath\space\endcsname
  {#1}{#2}}
\fi
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
%<latexrelease>\let\TextOrMath\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

```
%</2ekernel>
```

h a n g

13

Kernel Lengths

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed.
These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002
16
-2.5
-2.1
0.87865

A sign that my vision, rapid like all visions, if it had not lasted the space of an "amen," as the saying goes, had lasted almost the length of a "*Dies irae*."

Umbeco Eco, "The Name of the Rose", 1983

13.1 File n, lengths and the ltlength.dtx

This class defines a number of user commands for manipulating lengths. the code is straightforward. The `\newlength` command allocates a new internal skip register using the `\newskip` command from the allocations module.

`\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.
`\@settodim`	internal macro
`\@settopoint`	internal macro

`\newlength`{⟨*command name*⟩}

The `\newlength` is just syntactic sugar for PlainTeX's `\newskip`. The `\@ifdefinable` will produce an error message if the command has already been set.

```
\def\newlength#1{\@ifdefinable#1{\newskip#1}}
```

`\setlength`

```
\def\setlength#1#2{#1#2\relax}
```

`\addtolength`

```
\def\addtolength#1#2{\advance#1 #2\relax}
```

Example 1.1 Lengths

```
\bgroup
\newlength\alength
\setlength\alength{123.12pt}
\the\alength
```

```

\addtolength\alength{12.0pt}
\the\alength
\egroup
```

123.12pt
135.12pt

The \setto commands use a temporary box \@tempboxa to store the contents and then measure them using the internal macro \@settodim which boxes the contents and then uses TEX's ht andwd.

\@settodim{⟨⟩}{⟨⟩}{⟨⟩}

Sets a length to another

```
\def\@settodim#1#2#3{\setbox\@tempboxa\hbox{{#3}}#2#1\@tempboxa
%  Clear the memory afterwards (which might be a lot).
  \setbox\@tempboxa\box\voidb@x}
  \def\settodepth {\@settodim\dp}
```

\settoheight{⟨*dimen*⟩}

Sets the given length to the height of the box.

```
\def\settoheight{\@settodim\ht}
```

\settowidth {⟨*arg*⟩}{⟨*material*⟩}

Sets the width to be equal to the boxed content..

```
  \def\settowidth {\@settodim\wd}
```

\@settopoint{⟨*skip register*⟩}

The \@settopoint{⟨*skip register*⟩} 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 45.455pt when calulating a dimension. The method of rounding is interesting. Also it is interesting that this macro, is not used in the kernel at all, but is defined here for use with the standard classes (it is used to round off dimensions for page calculations).

Example 1.2 Rounding dimensions

```
\bgroup
  \makeatletter
  \def\@settopoint#1{\divide#1\p@\multiply#1\p@}
  \newlength\@test
  \setlength\@test{19.5pt}
  \@settopoint{\@test}
  \the\@test
  \makeatother
\egroup
```

19.0pt

13.2 Redefinitions and Extensions by Packages

The commands **??**[→P.??] and **??**[→P.??], **??**[→P.??] have been reimplemented by the package **calc**

to enable infix calculations.[1] The package also implements a number of other useful commands which are described below. The **pgf** prefixes all the commands, so it has no impact on the kernel commands which can still be used. This is a much better approach.

??[→P.??]	Declare #1 to be a new length command.
??[→P.??]	Set the length command, #1, to the value #2.
??[→P.??]	Increase the value of the length command, #1, by the value #2.
??[→P.??]	Set the length, #1 to the width of a box containing #2.
??[→P.??]	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.
`\widthof`{⟨*text*⟩}	Gets the natural width of ⟨*text*⟩.
`\heightof`{⟨*text*⟩}	
`\depthof`{⟨*text*⟩}	
`\maxof`	Select a maximum of two dimension expressions.
`\minof`	Select the minimum of two dimension expressions.
`\@settodim`	internal macro
`\@settopoint`	internal macro

[1] **calc**; **pkg-pgf**.

[2015/04/07 v3.1a LaTeX Kernel (NFSS Basic Macros)]

h a n g

14

The new font selection scheme for TEX macro packages

In this short file provided in the LATEX kernel the definitions for the basic commands for typesetting the logos for TEX, LaTeX and LATEX 2_ε are developed. These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002
16
-2.5
-2.1
0.87865

This file has version number ? dated ? The file credits It is currently mainatined by the LATEX Team. This file contains the main implementation of the 'low level' font selection commands. See other parts of the LATEX distribution, or *The LATEX Companion* for higher level documentation of the LATEX 'New' Font Selection Scheme.

> **Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

```
%<2ekernel>\expandafter\let\csname ↩
      ↪ver@autofss1.sty\endcsname\fmtversion
```

14.1 Preliminary macros

We define a number of macros that will be used later.

\@nomath

\@nomath is used by most macros that will have no effect in math mode. It issues a warning message.

```
<*2ekernel>
\def\@nomath#1{\relax\ifmmode
   \@font@warning{Command \noexpand#1invalid in math mode}\fi}
```

(*End definition for* `\@nomath`.)

```
\gdef\no@alphabet@error#1{\relax \ifmmode
    \@latex@error{Math\space alphabet\space identifier\space
          \noexpand#1is\space undefined\space in\space ↩
                ↪math\space
           version\space `\math@version'}%
        {Your\space requested\space math\space alphabet\space
         is\space undefined\space in\space the\space ↩
               ↪current\space
          math\space version.^^JCheck\space the\space ↩
                ↪spelling\space
          or\space use\space the\space ↩
                ↪\noexpand\SetMathAlphabet\space
          command.}
     \fi}
```

```
%\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@@n}
\let\mathgroup\fam
%\let\newfam\new@mathgroup
\@onlypreamble\new@mathgroup
```

14.2 Macros for setting up the tables

\DeclareFontShape {⟨*encoding*⟩}{⟨*family*⟩}{⟨*series*⟩}{⟨*shape*⟩}{⟨*loading-info*⟩} {⟨*loading-settings*⟩}

Declares a font shape combination; ⟨*loading-info*⟩ contains the information that combines sizes with external fonts. The syntax is complex and is described in detail in Section 4.3 of **fntguide**.

```
\DeclareFontShape{OT1}{cmr}{m}{sl}{%
<5-8> sub * cmr/m/n
<8> cmsl8
<9> cmsl9
<10> <10.95> cmsl10
<12> <14.4> <17.28> <20.74> <24.88> cmsl12
}{}
```

```
\def\DeclareFontShape{\begingroup
   \nfss@catcodes
```

We use `\expandafter \endgroup` to restore catcode in case something goes wrong with the argument parsing (suggested by Tim Van Zandt)

```
   \expandafter\endgroup
   \DeclareFontShape@}
```

Using internal macros with camel case variables, I am not too sure if it is good practice. I don't suggest you do this in your own packages.

`\DeclareFontShape@`

```
\def\DeclareFontShape@#1#2#3#4#5#6{%
   \expandafter\ifx\csname #1+#2\endcsname\relax
     \@latex@error{Font family `#1+#2' unknown}\@eha
   \else
     \expandafter
       \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
                                   \csname #5\endcsname}%
     \def\reserved@a{#6}%
     \global
     \expandafter\let\csname#5\expandafter\endcsname
        \ifx\reserved@a\@empty
          \@empty
        \else
          \reserved@a
        \fi
   \fi
  }
```

(*End definition for* `\DeclareFontShape@`.)

`\DeclareFixedFont` {⟨*cmd*⟩}{⟨*encoding*⟩}{⟨*family*⟩}{⟨*series*⟩}{⟨*shape*⟩}{⟨*size*⟩}

Define a direct font switch that avoids all overhead. Declares command hcmdi to be a font switch which selects the font that is specified by the attributes hencodingi, hfamilyi, hseriesi, hshapei, and hsizei.
The font is selected without any adjustments to baselineskip and other surrounding conditions.
This example makes `{\picturechar .}` select a small dot very quickly:
`\DeclareFixedFont{\picturechar}{OT1}{cmr}{m}{n}{5}`

```
\def\DeclareFixedFont#1#2#3#4#5#6{%
   \begingroup
      \math@fontsfalse
      \every@math@size{}%
      \fontsize{#6}\z@
      \usefont{#2}{#3}{#4}{#5}%
      \global\expandafter\let\expandafter#1\the\font
   \endgroup
  }
```

```
\def\do@subst@correction{%
       \xdef\subst@correction{%
          \font@name
          \global\expandafter\font
            \csname \curr@fontshape/\f@size\endcsname
            \noexpand\fontname\font
           \relax}%
```

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

```
       \aftergroup\subst@correction
}
```

`\DeclareFontFamily` {⟨*encoding*⟩}{⟨*family*⟩}{⟨*loading-settings*⟩}

Declares a font family ⟨*family*⟩ to be available in encoding scheme ⟨*encoding*⟩. The ⟨*loading-settings*⟩ are executed immediately after loading any font with this encoding and family.
Checks that ⟨*encoding*⟩ was previously declared. This example refers to the Computer Modern Typewriter font family in the Cork encoding:
`\DeclareFontFamily{T1}{cmtt}{\hyphenchar\font=-1}`
Each `.fd` file should contain exactly one `\DeclareFontFamily` combination.

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

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

```
% \@tempswafalse
% \def\reserved@b{#1}%
% \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
%      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
% \cdp@list
% \if@tempswa
 \@ifundefined{T@#1}%
    {%
     \@latex@error{Encoding scheme  `#1' unknown}\@eha
    }%
    {%
```

Now we have to define the macro `\`⟨*#1*⟩`+`⟨*#2*⟩ to contain `#3`. But since most of the time `#3` will be empty we use `\let` in a tricky way rather than a simple `\def` since this will save internal memory. We store the argument `#3` in a temporary macro `\reserved@a`.

```
     \def\reserved@a{#3}%
```

We compare **\reserved@a** with **\@empty** If these two are the same we **\let** the 'extra' macro equal to **\@empty** which is not the same a doing a **\let** to **\reserved@a** — the latter would blow one extra memory location rather then reusing the one from **\@empty**.

```
  \global
  \expandafter\let\csname #1+#2\expandafter\endcsname
         \ifx \reserved@a\@empty
           \@empty
         \else \reserved@a
         \fi
 }%
}
```

```
\let\cdp@list\@empty
\@onlypreamble\cdp@list
```

```
\let\cdp@elt\relax
\@onlypreamble\cdp@elt
```

\DeclareFontEncoding {⟨*encoding*⟩}{⟨*text-settings*⟩}{⟨*math-settings*⟩}

This function declares a new font encoding scheme.
The ⟨*text-settings*⟩ are declarations which are executed every time **\selectfont** changes the encoding to be ⟨*encoding*⟩.
The ⟨*math-settings*⟩ are similar but are for math alphabets. They are executed whenever a math alphabet with this encoding is called.
The command also saves the value of ⟨*encoding*⟩ in the macro **??**→P. **??**.
Spaces within the arguments are ignored to avoid surplus spaces in the document.[1]

```
\def\DeclareFontEncoding{%
```

First we start with ignoring all blanks and newlines since every surplus space in the second or third argument will come out in a weird place in the document.

```
  \begingroup
  \nfss@catcodes
  \expandafter\endgroup
  \DeclareFontEncoding@}
\@onlypreamble\DeclareFontEncoding
```

```
\def\DeclareFontEncoding@#1#2#3{%
  \expandafter
  \ifx\csname T@#1\endcsname\relax
     \def\cdp@elt{\noexpand\cdp@elt}%
     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
                    {\default@family}{\default@series}%
                    {\default@shape}}%
```

To support encoding dependent commands (like accents) we initialise the command

```
     \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
  \else
     \@font@info{Redeclaring font encoding #1}%
  \fi
```

[1] **fntguide**.

```
  \global\@namedef{T@#1}{#2}%
  \global\@namedef{M@#1}{\default@M#3}%
```

Keep a record of the last encoding being declared:

```
  \xdef\LastDeclaredEncoding{#1}%
  }
\@onlypreamble\DeclareFontEncoding@
```

\LastDeclaredEncoding

```
\def\LastDeclaredEncoding{}
```

\DeclareFontSubstitution {⟨*encoding*⟩}{⟨*family*⟩}{⟨*series*⟩}{⟨*shape*⟩}

Declares the default values for font substitution which will be used when a font with encoding hencodingi should be loaded but no font can be found with the current attributes.

These substitutions are local to the encoding scheme because the encoding scheme is never substituted! They are tried in the order ⟨*shape*⟩ then ⟨*series*⟩ and finally ⟨*family*⟩.

If no defaults are set up for an encoding, the values given by **??**→P.?? are used.

The font specification for hencodingihfamilyihseriesihshapei must have been de- fined by **??**→P.?? before the **\begin{document}** is reached.

Example:

```
\DeclareFontSubstitution{T1}{cmr}{m}{n}
```

```
\def\DeclareFontSubstitution#1#2#3#4{%
  \expandafter
  \ifx\csname T@#1\endcsname\relax
    \@latex@error{Encoding scheme  `#1' unknown}\@eha
  \else
    \begingroup
       \edef\reserved@a{#1}%
       \toks@{}%
       \def\cdp@elt##1##2##3##4{%
          \def\reserved@b{##1}%
          \ifx\reserved@a\reserved@b
             \addto@hook\toks@{\cdp@elt{##1}{#2}{#3}{#4}}%
          \else
             \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
          \fi}%
        \cdp@list
        \xdef\cdp@list{\the\toks@}%
    \endgroup
    \global
    \@namedef{D@#1}{%
           \def\default@family{#2}%
           \def\default@series{#3}%
           \def\default@shape{#4}%
           }%
```

```
  \fi
 }
\@onlypreamble\DeclareFontSubstitution
```

\DeclareFontEncodingDefaults

```
\def\DeclareFontEncodingDefaults#1#2{%
  \ifx\relax#1\else
    \ifx\default@T\@empty\else
      \@font@info{Overwriting encoding scheme text defaults}%
    \fi
    \gdef\default@T{#1}%
  \fi
  \ifx\relax#2\else
    \ifx\default@M\@empty\else
      \@font@info{Overwriting encoding scheme math defaults}%
    \fi
    \gdef\default@M{#2}%
  \fi
}
\@onlypreamble\DeclareFontEncodingDefaults
```

```
\let\default@T\@empty
\let\default@M\@empty
```

\DeclarePreloadSizes

```
\def\DeclarePreloadSizes#1#2#3#4#5{%
 \@ifundefined{T@#1}%
   {\@latex@error{Encoding scheme  `#1' unknown}\@eha}%
   {%
   \begingroup
    \def\reserved@f##1,{%
        \if>##1>%
          \let\reserved@f\relax
        \else
          \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
          \pickup@font
          \global\expandafter\let\font@name\relax
        \fi
        \reserved@f}%
     \reserved@f#5,,%
   \endgroup
   }%
}
\@onlypreamble\DeclarePreloadSizes
```

```
\newif\ifmath@fonts \math@fontstrue
```

\DeclareMathSizes

```
\def\DeclareMathSizes{%
  \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
          {\@DeclareMathSizes{}}}
\@onlypreamble\DeclareMathSizes

%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\@DeclareMathSizes}%
%<latexrelease>                 {Arbitrary units in \DeclareMathSizes}%
%<*2ekernel|latexrelease>
\def\@DeclareMathSizes #1#2#3#4#5{%
  \@defaultunits\dimen@ #2pt\relax\@nnil
  \if $#3$%
    \expandafter\let\csname S@\strip@pt\dimen@\endcsname\math@fontsfalse
  \else
    \@defaultunits\dimen@ii #3pt\relax\@nnil
    \@defaultunits\@tempdima #4pt\relax\@nnil
    \@defaultunits\@tempdimb #5pt\relax\@nnil
    \toks@{#1}%
    \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
      \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
      \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
      \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
      \the\toks@
    }%
  \fi
}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\@DeclareMathSizes}%
%<latexrelease>                 {Arbitrary units in \DeclareMathSizes}%
%<latexrelease>\def\@DeclareMathSizes#1#2#3#4#5{%
%<latexrelease>    \@defaultunits\dimen@#2pt\relax\@nnil
%<latexrelease>    \if$#3$%
%<latexrelease>      \expandafter \let
%<latexrelease>        \csname S@\strip@pt\dimen@\endcsname
%<latexrelease>        \math@fontsfalse
%<latexrelease>    \else
%<latexrelease>      \expandafter \gdef
%<latexrelease>      \csname S@\strip@pt\dimen@\endcsname
%<latexrelease>            {\gdef\tf@size{#3}\gdef\sf@size{#4}%
%<latexrelease>                              \gdef\ssf@size{#5}%
%<latexrelease>             #1%
%<latexrelease>                              }%
%<latexrelease>    \fi}%
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
\@onlypreamble\@DeclareMathSizes
```

\fontencoding
\f@encoding

As we said in the introduction a font is described by four parameters. We first

define macros to specify the wanted *family*, *series*, or *shape*. These are simply recorded in internal macros **\f@family**, **\f@series**, and **\f@shape**, resp. We use **\edef**'s so that the arguments can also be macros.

```
\DeclareRobustCommand\fontencoding[1]{%
    \expandafter\ifx\csname T@#1\endcsname\relax
      \@latex@error{Encoding scheme `#1' unknown}\@eha
    \else
      \edef\f@encoding{#1}%
      \ifx\cf@encoding\f@encoding
```

If the new encoding is the same as the old encoding we have nothing to do. However, in case we had a sequence of several encoding changes without a **\selectfont** in-between we can save processing by making sure that **\enc@update** is **\relax**.

```
        \let\enc@update\relax
      \else
```

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

```
        \let\enc@update\@@enc@update
      \fi
    \fi
}
```

(*End definition for* `\fontencoding` *and* `\f@encoding`.)

```
\def\@@enc@update{%
          \expandafter
          \let
            \csname\cf@encoding -cmd\endcsname
            \@changed@cmd
          \expandafter
          \let
            \csname\f@encoding-cmd\endcsname
            \@current@cmd
        \default@T
        \csname T@\f@encoding\endcsname
        \csname D@\f@encoding\endcsname
        \let\enc@update\relax
        \let\cf@encoding\f@encoding
}
\let\enc@update\relax
\DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
```

\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.

```
\def\usefont#1#2#3#4{\fontencoding{#1}\fontfamily{#2}%
             \fontseries{#3}\fontshape{#4}\selectfont
             \ignorespaces}
\DeclareRobustCommand\linespread[1]
   {\set@fontsize{#1}\f@size\f@baselineskip}
\DeclareRobustCommand\fontsize[2]
   {\set@fontsize\baselinestretch{#1}{#2}}
\let\f@family\@empty
\let\f@series\@empty
\let\f@shape\@empty
\let\f@size\@empty
\let\f@baselineskip\@empty
\let\f@linespread\@empty
\let\f@encoding\@empty
\let\cf@encoding\@empty
\def\@defaultunits{\afterassignment\remove@to@nnil}
```

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

```
\begingroup
  \catcode`P=12
  \catcode`T=12
  \lowercase{
    \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}}
  \expandafter\endgroup\x
\def\strip@pt{\expandafter\rem@pt\the}
```

(*End definition for* `\strip@pt` *and* `\rem@pt`.)

\mathversion {⟨*version*⟩}

This command selects a math version; it can only be used outside math mode. For example, **\boldmath** is defined to be **\mathversion{bold}**.

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

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

```
          \gdef\glb@currsize{}%
```

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

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

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

Anyway, since we don't have this and won't have it in LATEX 2_ε we need to implement it in a somewhat slower way.

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

\frozen@everymath
\frozen@everydisplay

New internal names for **\everymath** and **\everydisplay**.

```
\let\frozen@everymath\everymath
\let\frozen@everydisplay\everydisplay
```

(*End definition for* `\frozen@everymath` *and* `\frozen@everydisplay`.)

\everymath
\everydisplay

```
\newtoks\everymath
\newtoks\everydisplay
```

(*End definition for* `\everydisplay`.)

```
\frozen@everymath = {\check@mathfonts
                     \the\everymath}
\frozen@everydisplay = {\check@mathfonts
                        \the\everydisplay}
```

\curr@math@size

This holds locally the current math size.

```
\let\curr@math@size\@empty
```

(*End definition for* `\curr@math@size`.)

14.3 Macros for loading fonts

\pickup@font

The macro **\pickup@font** which is used in **\selectfont** is very simple: if the font name is undefined (i.e. not known yet) it calls **\define@newfont** to load it.

```
\def\pickup@font{%
    \expandafter \ifx \font@name \relax
       \define@newfont
    \fi}
```

\split@name

\pickup@font assumes that \font@name is set but it is sometimes called when \f@family, \f@series, \f@shape, or \f@size may have the wrong settings (see, e.g., the definition of \getanddefine@fonts). Therefore we need a macro to extract font *family*, *series*, *shape*, and *size* from the font name. To this end we define \split@name which takes the font name as a list of characters of \catcode 12 (without the backslash at the beginning) delimited by the special control sequence \@nil. This is not very complicated: we first ensure that / has the right \catcode

```
{\catcode`\/=12
```

and define \split@name so that it will define our private \f@encoding, \f@family, \f@series, \f@shape, and \f@size macros.

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

\curr@fontshape

Abbreviation which may get removed again for speed.

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

Example 3.1 Currentfontshape

```
\makeatletter
\curr@fontshape
\makeatother
```

TU/Georgia(0)/m/n

\define@newfont

Now we can tackle the problem of defining a new font.

```
\def\define@newfont{%
```

We have already mentioned that the token list that \split@name will get as argument must not start with a backslash. To reach this goal we will set the \escapechar to −1 so that the \string primitive will not generate an escape character. To keep this change local we open a group. We use \begingroup for this purpose since \define@newfont might be called in math mode, and an empty \bgroup...\egroup would add an empty Ord atom to the math list and thus affect the spacing.

Also locally redefine **\typeout** so that 'No file ...fd' Warnings become Font Info message just sent to the log file.

```
\begingroup
 \let\typeout\@font@info
 \escapechar\m@ne
```

Then we extract *encoding scheme*, *family*, *series*, *shape*, and *size* from the font name. Note the four **\expandafter**'s so that **\font@name** is expanded first, then **\string**, and finally **\split@name**.

```
 \expandafter\expandafter\expandafter
    \split@name\expandafter\string\font@name\@nil
```

If the **\curr@fontshape** combination is not available, (i.e. undefined) we call the macro **\wrong@fontshape** to take care of this case. Otherwise **\extract@font** will load the external font for us.

```
 \expandafter\ifx
     \csname\curr@fontshape\endcsname \relax
   \try@load@fontshape % try always
 \fi
 \expandafter\ifx
    \csname\curr@fontshape\endcsname \relax
   \wrong@fontshape\else
```

To allow substitution we call the **curr@fontshape** macro which usually will expand to **\relax** but may hold code for substitution (see **\subst@fontshape** definition).

```
   \csname\curr@fontshape\endcsname
   \extract@font\fi
```

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

```
\endgroup}
```

```
\def\try@load@fontshape{%
  \expandafter
  \ifx\csname \f@encoding+\f@family\endcsname\relax
    \@font@info{Try loading font information for
                  \f@encoding+\f@family}%
   \global\expandafter\let
      \csname\f@encoding+\f@family\endcsname\@empty
    \nfss@catcodes
    \let\nfss@catcodes\relax
    \edef\reserved@a{%
      \lowercase{%
        \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}}%
    \reserved@a\relax
         {\@input@{\f@encoding\f@family.fd}}%
  \fi}
```

\nfss@catcodes

This macro should contain the standard **\catcode** assignments to all characters which are used in the commands found in an `.fd` file and which

might have special **\catcode**s in the middle of a document. If necessary, this list can be extended in a package file using a suitable number of **\expandafter**, i.e.,

```
\expandafter\def\expandafter\nfss@catcodes
      \expandafter{\nfss@catcodes <additional settings>}
```

Note, that this macro might get executed several times since it is also called by **\DeclareFontShape**, thus it probably should not be misused as a general purpose hook.

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

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

```
  \makeatletter
  \catcode`\ 9%
  \catcode`\^^I9%
  \catcode`\^^M9%
```

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

```
  \catcode`\\\z@
  \catcode`\{\@ne
  \catcode`\}\tw@
  \catcode`\#6%
  \catcode`\^7%
  \catcode`\%14%
```

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

```
 \@makeother\<%
 \@makeother\>%
 \@makeother\*%
 \@makeother\.%
 \@makeother\-%
 \@makeother\/%
 \@makeother\[%
 \@makeother\]%
 \@makeother\`%
 \@makeother\'%
 \@makeother\"%
}
```

\DeclareErrorFont

Declare the last resort shape! We assume that in this fontshape there is a 10pt font but it doesn't really matter. We only loose one macro name if the assumption is false. But at least the font should be there!

```
\def\DeclareErrorFont#1#2#3#4#5{%
    \xdef\error@fontshape{%
        \noexpand\expandafter\noexpand\split@name\noexpand\string
        \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
        \noexpand\@nil}%
%     \gdef\f@encoding{#1}%
    \gdef\default@family{#2}%
    \gdef\default@series{#3}%
```

```
      \gdef\default@shape{#4}%
      \global\let\f@family\default@family
      \global\let\f@series\default@series
      \global\let\f@shape\default@shape
      \gdef\f@size{#5}%
      \gdef\f@baselineskip{#5pt}%
}
\@onlypreamble\DeclareErrorFont

%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
%<latexrelease>                 {Font substitution in preamble}%
%<*2ekernel|latexrelease>
\def\wrong@fontshape{%
    \csname D@\f@encoding\endcsname      % install defaults if in math
    \edef\reserved@a{\csname\curr@fontshape\endcsname}%
  \ifx\last@fontshape\reserved@a
     \errmessage{Corrupted NFSS tables}%
     \error@fontshape
  \else
    \let\f@shape\default@shape
    \expandafter\ifx\csname\curr@fontshape\endcsname\relax
       \let\f@series\default@series
        \expandafter
          \ifx\csname\curr@fontshape\endcsname\relax
           \let\f@family\default@family
           \begingroup
              \try@load@fontshape
           \endgroup
        \fi \fi
  \fi
     \@font@warning{Font shape `\expandafter\string\reserved@a'
                     \expandafter\@gobble\string\@undefined\MessageBreak
                   using `\curr@fontshape' instead\@wrong@font@char}%
    \global\let\last@fontshape\reserved@a
    \gdef\@defaultsubs{%
      \@font@warning{Some font shapes were not available, defaults
                        substituted.\@gobbletwo}}%
    \global\expandafter\expandafter\expandafter\let
       \expandafter\reserved@a
           \csname\curr@fontshape\endcsname
    \xdef\font@name{%
      \csname\curr@fontshape/\f@size\endcsname}%
```

```
    \pickup@font}
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
%<latexrelease>                 {Font substituation in preamble}%
%<latexrelease>\def\wrong@fontshape{%
%<latexrelease>    \csname D@\f@encoding\endcsname      % install defaults if in math
%<latexrelease>    \edef\reserved@a{\csname\curr@fontshape\endcsname}%
%<latexrelease>  \ifx\last@fontshape\reserved@a
%<latexrelease>     \errmessage{Corrupted NFSS tables}%
%<latexrelease>     \error@fontshape
%<latexrelease>  \else
%<latexrelease>    \let\f@shape\default@shape
%<latexrelease>    \expandafter\ifx\csname\curr@fontshape\endcsname\relax
%<latexrelease>       \let\f@series\default@series
%<latexrelease>        \expandafter
%<latexrelease>          \ifx\csname\curr@fontshape\endcsname\relax
%<latexrelease>           \let\f@family\default@family
%<latexrelease>        \fi \fi
%<latexrelease>  \fi
%<latexrelease>     \@font@warning{Font shape `\expandafter\string\reserved@a'
%<latexrelease>                    \expandafter\@gobble\string\@undefined\MessageBreak
%<latexrelease>                   using `\curr@fontshape' instead\@wrong@font@char}%
%<latexrelease>    \global\let\last@fontshape\reserved@a
%<latexrelease>    \gdef\@defaultsubs{%
%<latexrelease>      \@font@warning{Some font shapes were not available, defaults
%<latexrelease>                     substituted.\@gobbletwo}}%
%<latexrelease>    \global\expandafter\expandafter\expandafter\let
%<latexrelease>       \expandafter\reserved@a
%<latexrelease>           \csname\curr@fontshape\endcsname
%<latexrelease>    \xdef\font@name{%
%<latexrelease>      \csname\curr@fontshape/\f@size\endcsname}%
%<latexrelease>    \pickup@font}
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
\let\@wrong@font@char\@empty
\let\@defaultsubs\relax
\def\strip@prefix#1>{}
\let\install@mathalphabet\gdef
\let\math@fonts\@empty
%\def\select@group#1#2#3{\relax\ifmmode
```

```
%  \begingroup
%     \escapechar\m@ne
%    \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
%    \globaldefs\@ne  \math@fonts
%  \endgroup
%  \xdef#1{\noexpand\use@mathgroup\noexpand#2%
%          {\number\csname c@mv@\math@version\endcsname}}%
%     \expandafter\extract@alph@from@version
%        \csname mv@\math@version\expandafter\endcsname
%       \expandafter{\number\csname c@mv@\math@version\endcsname}%
%       #1%
%     \stepcounter{mv@\math@version}%
%\expandafter #1\fi}
```

\extract@alph@from@version

```
\def\extract@alph@from@version#1#2#3{%
    \def\reserved@a##1\install@mathalphabet#3##2##3\@nil{%
        \def\reserved@b{##2}%
        \def\reserved@c####1{\gdef#1{##1####1##3}}}%
    \expandafter\reserved@a#1\@nil
    \def\reserved@a\select@group#3##1##2\@nil{%
```

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

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

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

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

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

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

(*End definition for* `\extract@alph@from@version`.)

\math@bgroup
\math@egroup

```
\let\math@bgroup\bgroup
\def\math@egroup#1{#1\egroup}
```

(*End 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.

```
\gdef\calculate@math@sizes{%
  \@font@info{Calculating\space math\space sizes\space for\space
              size\space <\f@size>}%
  \dimen@\f@size \p@
  \@tempdimb \defaultscriptratio \dimen@
  \dimen@ \defaultscriptscriptratio \dimen@
  \expandafter\xdef\csname S@\f@size\endcsname{%
    \gdef\noexpand\tf@size{\f@size}%
    \gdef\noexpand\sf@size{\strip@pt\@tempdimb}%
    \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
    \noexpand\math@fontstrue}}
```

(*End definition for* `\calculate@math@sizes`.)

\defaultscriptratio
ultscriptscriptratio

The default ratio for math sizes is:
1 to **\defaultscriptratio** to **\defaultscriptscriptratio**.
By default this is 1 to .7 to .5.

```
\def\defaultscriptratio{.7}
\def\defaultscriptscriptratio{.5}
```

(*End definition for* `\defaultscriptratio` *and* `\defaultscriptscriptratio`.)

```
\ifx\noaccents@\@undefined
  \let\noaccents@\@empty
\fi
```

\showhyphens

The **\showhyphens** command must be redefined since the version in `plain.tex` uses **\tenrm**. We have also made some further adjustments for its use in LaTeX.

```
\gdef\showhyphens#1{%
  \setbox0\vbox{%
    \color@begingroup
    \everypar{}%
    \parfillskip\z@skip\hsize\maxdimen
    \normalfont
    \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
    \color@endgroup}}
```

\addto@hook

We need a macro to add tokens to a hook.

```
\long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}
```

\@vpt

```
 \def\@vpt{5}
```

\@vipt

```
\def\@vipt{6}
```

\@viipt

```
\def\@viipt{7}
```

\@viiipt

```
\def\@viiipt{8}
```

\@ixpt

```
\def\@ixpt{9}
```

\@xpt

```
\def\@xpt{10}
```

\@xipt

```
\def\@xipt{10.95}
```

\@xiipt

```
\def\@xiipt{12}
```

\@xivpt

```
\def\@xivpt{14.4}
```

\@xviipt

```
\def\@xviipt{17.28}
```

\@xxpt

```
\def\@xxpt{20.74}
```

\@xxvpt

```
\def\@xxvpt{24.88}
```

```
%</2ekernel>
```

h a n g

15

ltfsstrc

In this short file provided in the LaTeX kernel the definitions for the basic commands for typesetting the logos for TeX, LaTeX and LaTeX 2_ε are developed. These can be typeset using the macros `\TeX`, `\LaTeX` and `\LaTeXe`. 21.00002

16

-2.5

-2.1

0.87865

This package contains the code for tracing font loading and font changes. It basically overlays some of the low-level functions of NFSS with additional code used for tracing.

The package accepts the following options:

errorshow Write all information about font changes etc. only to the transcript file unless an error happens. This means that information about font substitution will not be shown on the terminal.

warningshow Show all NFSS warnings on the terminal. This setting corresponds to the default behaviour of NFSS if the `tracefnt` package is *not* loaded!

infoshow Show all NFSS warning and all NFSS info messages (that are normally only written to the transcript file) also on the terminal. This is the default if the `tracefnt` package is loaded.

debugshow In addition to `infoshow` show also changing of math fonts as far as possible (this option can produce a large amount of output.

loading Show the name of external fonts when they are loaded. This option shows only "newly" loaded fonts not those already preloaded in the format or the class file before the `tracefnt` package became active.

pausing Turn all font warnings into errors so that LaTeX will stop.

15.1 A driver for this document

15.2 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.

```
%<*package>
%\NeedsTeXFormat{LaTeX2e}
%\ProvidesPackage{tracefnt}[??/??/?? v?.??
%                            Standard LaTeX package (font
      (tracing)]
%</package>
```

The `debug` module makes use of commands contained in a special package file named `trace.sty`.[1]

```
%<+debug> \input trace.sty
```

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

15.3 Handling Options

\tracingfonts

Here is the definition of the integer register for the font trace. As a default in a package file we use 1 to give error messages if fonts are substituted. If this code is used for debugging or tracing reasons in the format file (i.e. in `fam.dtx`) we use 0 as the default. But if no font trace is used we build a definition that will produce a warning message.

```
%<*2ekernel>
\def\tracingfonts{%
  \@font@warning{Command \noexpand\tracingfonts
           not provided.\MessageBreak
           Use the `tracefnt' package.\MessageBreak Command ↱
                ↳found:}%
       \count@}
%</2ekernel>
```

The `\count@` in the line above will remove the number after `\tracingfonts`. Note that this definition will be overwritten be the next line if one of these modules are included.

```
%<*package,trace,debug>
\newcount\tracingfonts
\tracingfonts=0
%</package,trace,debug>
```

(*End definition for* `\tracingfonts`.)

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

```
%<*package>
\DeclareOption{errorshow}{%
   \def\@font@info#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
    \def\@font@warning#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                        {LaTeX Font Warning: #1}}%
     }
\DeclareOption{warningshow}{%
   \def\@font@info#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
    \def\@font@warning#1{%
         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                        {LaTeX Font Warning: #1}}%
     }
\DeclareOption{infoshow}{%
   \def\@font@info#1{%
         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
    \def\@font@warning#1{%
```

```
          \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                          {LaTeX Font Warning: #1}}%
      }
\DeclareOption{loading}{%
    \tracingfonts\tw@
   }
\DeclareOption{debugshow}{%
    \ExecuteOptions{infoshow}%
    \tracingfonts\thr@@
   }
\DeclareOption{pausing}{%
    \def\@font@warning#1{%
      \GenericError
             {(Font)\@spaces\@spaces\@spaces\space\space}%
             {LaTeX Font Warning: #1}%
             {See the LaTeX Companion for details.}%
             {I'll stop for every LaTeX Font Warning because
              you requested\MessageBreak the `pausing' option
              to the tracefnt package.}}%
   }
\ExecuteOptions{infoshow}
\ProcessOptions
%</package>
```

We also need a default definition inside the kernel:

```
%<*2ekernel>
\def\@font@info#1{%
          \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                       {LaTeX Font Info: \space\space\space#1}}%
\def\@font@warning#1{%
          \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                          {LaTeX Font Warning: #1}}%
%</2ekernel>
```

15.3.1 General font loading

```
%<*2ekernel|package>
\def\extract@font{%
   \get@external@font
   \global\expandafter\font\font@name\external@font\relax
%<*trace>
    \ifnum \tracingfonts >\@ne
    \@font@info{External font `\external@font'
                loaded as\MessageBreak \font@name}\fi
%</trace>
    \font@name \relax
    \csname \f@encoding+\f@family\endcsname
    \csname\curr@fontshape\endcsname
```

```
    \relax
       }
%</2ekernel|package>
%<*2ekernel>
\def\get@external@font{%
   \let\external@font\@empty
   \edef\font@info{\expandafter\expandafter\expandafter\string
        \csname \curr@fontshape \endcsname}%
   \try@size@range
   \ifx\external@font\@empty
      \try@size@substitution
      \ifx\external@font\@empty
         \@latex@error{Font \expandafter ↩
               ↪\string\font@name\space
                     not found}\@eha
         \error@fontshape
         \get@external@font
   \fi\fi
}
%</2ekernel>
%<*2ekernel|package>
\DeclareRobustCommand\selectfont
        {%
%<+debug>  \pushtracing
%<+debug>  \ifnum\tracingfonts<4 \tracingoff
%<+debug>  \else \tracingon\p@selectfont \fi
    \ifx\f@linespread\baselinestretch \else
      \set@fontsize\baselinestretch\f@size\f@baselineskip \fi
    \xdef\font@name{%
      \csname\curr@fontshape/\f@size\endcsname}%
    \pickup@font
    \font@name
%<*trace>
    \ifnum \tracingfonts>\tw@
      \@font@info{Switching to \font@name}\fi
%</trace>
    \size@update
    \enc@update
%<+debug> \poptracing
    }
\def\set@fontsize#1#2#3{%
    \@defaultunits\@tempdimb#2pt\relax\@nnil
    \edef\f@size{\strip@pt\@tempdimb}%
    \@defaultunits\@tempskipa#3pt\relax\@nnil
    \edef\f@baselineskip{\the\@tempskipa}%
    \edef\f@linespread{#1}%
```

```
    \let\baselinestretch\f@linespread
      \def\size@update{%
        \baselineskip\f@baselineskip\relax
        \baselineskip\f@linespread\baselineskip
        \normalbaselineskip\baselineskip
        \setbox\strutbox\hbox{%
          \vrule\@height.7\baselineskip
                \@depth.3\baselineskip
                \@width\z@}%
%<*trace>
   \ifnum \tracingfonts>\tw@
      \ifx\f@linespread\@empty
        \let\reserved@a\@empty
      \else
        \def\reserved@a{\f@linespread x}%
      \fi
      \@font@info{Changing size to \f@size/\reserved@a
                \f@baselineskip}%
      \aftergroup\type@restoreinfo \fi
%</trace>
        \let\size@update\relax}%
  }
\let\size@update\relax
%<*trace>
   \def\type@restoreinfo{%
      \ifx\f@linespread\@empty
        \let\reserved@a\@empty
      \else
        \def\reserved@a{\f@linespread x}%
      \fi
      \@font@info{Restoring size to
                \f@size/\reserved@a\f@baselineskip}}
%</trace>
\def\glb@settings{%
     \expandafter\ifx\csname S@\f@size\endcsname\relax
       \calculate@math@sizes
     \fi
     \csname S@\f@size\endcsname
     \ifmath@fonts
%<*trace>
       \ifnum \tracingfonts>\tw@
         \@font@info{Setting up math fonts for
                     \f@size/\f@baselineskip}\fi
%</trace>
       \begingroup
         \escapechar\m@ne
         \csname mv@\math@version \endcsname
```

```
          \globaldefs\@ne
          \math@fonts
          \let \glb@currsize \f@size
        \endgroup
         \the\every@math@size
%<*trace>
        \else
          \ifnum \tracingfonts>\tw@
            \@font@info{No math setup for
                        \f@size/\f@baselineskip}\fi
%</trace>
      \fi
}
%</2ekernel|package>
%<*2ekernel>
\def\baselinestretch{1}
\newtoks\every@math@size
\every@math@size={}
%</2ekernel>
%<*2ekernel|package>
\def\check@mathfonts{%
  \ifx \glb@currsize \f@size
%<*trace>
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: no change \f@size\space
         curr/global (\curr@math@size/\glb@currsize)}\fi
%</trace>
  \else
%<*trace>
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: setting up \f@size\space
         curr/global (\curr@math@size/\glb@currsize)}\fi
%</trace>
     \glb@settings
     \init@restore@glb@settings
  \fi
  \let\curr@math@size\f@size
  \def\init@restore@glb@settings{\aftergroup\restglb@settings}%
}
%<-trace>\let\init@restore@glb@settings\relax
%<*trace>
\def\init@restore@glb@settings{%
       \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: no resetting (not in
                    nested math)}\fi
}
%</trace>
\def\restglb@settings{%
%<*trace>
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: restoring}\fi
```

```
%</trace>
      \begingroup
        \let\f@size\curr@math@size
        \ifx\glb@currsize \f@size
%<*trace>
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: ... already okay (\f@size)}\fi
%</trace>
        \else
%<*trace>
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: ... to \f@size}\fi
%</trace>
          \glb@settings
        \fi
      \endgroup
}
\def\use@mathgroup#1#2{\relax\ifmmode
%<*trace>
  \ifnum \tracingfonts>\tw@
    \count@#2\relax
    \@font@info{Using \noexpand\mathgroup
              (\the\count@) #2}\fi
%</trace>
     \math@bgroup
         \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
         #1\fi
         \mathgroup#2\relax
     \expandafter\math@egroup\fi}%
%<*trace>
  \ifx\math@bgroup\bgroup
    \def\math@egroup#1{#1\egroup
      \ifnum \tracingfonts>\tw@
     \@font@info{Restoring \noexpand\mathgroup
        (\ifnum\mathgroup=\m@ne default\else \the\mathgroup ⤸
             ⤹\fi)%
        }\fi}
  \fi
%</trace>
\def\getanddefine@fonts#1#2{%
%<+debug>  \pushtracing
%<+debug>  \ifnum\tracingfonts<4 \tracingoff
%<+debug>  \else \tracingon\getanddefine@fonts \fi
%<*trace>
  \ifnum \tracingfonts>\tw@
  \count@#1\relax
    \@font@info{\noexpand\mathgroup (\the\count@) #1 ⤸
          ⤹:=\MessageBreak
               \string#2 \tf@size/\sf@size/\ssf@size}\fi
%</trace>
```

```
  \xdef\font@name{\csname \string#2/\tf@size\endcsname}%
  \pickup@font \let\textfont@name\font@name
  \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
  \pickup@font \let\scriptfont@name\font@name
  \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
  \pickup@font
  \edef\math@fonts{\math@fonts
                \textfont#1\textfont@name
                \scriptfont#1\scriptfont@name
                \scriptscriptfont#1\font@name}%
%<+debug> \poptracing
    }
%</2ekernel|package>
%<*2ekernel>
\def\ifnot@nil#1{\def\reserved@a{#1}%
  \ifx\reserved@a\@nnil \expandafter\@gobble
  \else \expandafter\@firstofone\fi}
\def\remove@to@nnil#1\@nnil{}
\def\remove@angles#1>{\set@simple@size@args}
\def\remove@star#1*{#1}
\def\extract@sizefn#1*#2\@nil{%
  \if>#2>\set@size@funct@args#1\@nil
         \let\sizefn@info\@empty
  \else\expandafter\set@size@funct@args\remove@star#2\@nil
       \def\sizefn@info{#1}\fi
  }
% % this could be replaced by \try@size@range making the ↩
      ↪subst slower!
\def\try@simple@size{%
    \def\reserved@a{\def\extract@fontinfo####1}%
    \expandafter\reserved@a\expandafter<\f@size>##2<##3\@nnil{%
          \ifnot@nil{##2}%
            {\set@simple@size@args##2<##3\@nnil
             \execute@size@function\sizefn@info
            }}%
    \expandafter\expandafter
    \expandafter\extract@fontinfo\expandafter\font@info
    \expandafter<\f@size>\@nil<\@nnil
}
\def\set@simple@size@args#1<{%
          \if<#1<%
            \expandafter\remove@angles
          \else
            \extract@sizefn#1*\@nil
            \expandafter\remove@to@nnil
          \fi}
```

```
\def\extract@rangefontinfo#1<#2>{%
        \is@range#2->\@nil#2>}
\def\is@range#1-#2\@nil{%
   \if>#2\expandafter\check@single\else
      \expandafter\check@range\fi}
\def\check@range#1-#2>#3<#4\@nnil{%
  \ifnot@nil{#3}{%
     \def\reserved@f{\extract@rangefontinfo<#4\@nnil}%
     \upper@bound0#2\p@
     \ifdim\upper@bound=\z@ \upper@bound\maxdimen\fi
     \ifdim \f@size \p@<\upper@bound
       \lower@bound0#1\p@
       \ifdim \f@size \p@<\lower@bound
        \else
          \set@simple@size@args#3<#4\@nnil
          \execute@size@function\sizefn@info
         \ifx\external@font\@empty
         \else
           \let\reserved@f\@empty
         \fi
       \fi
     \fi
     \reserved@f}}
\newdimen\lower@bound
\newdimen\upper@bound
\def\check@single#1>#2<#3\@nnil{%
     \def\reserved@f{\extract@rangefontinfo<#3\@nnil}%
     \ifdim \f@size \p@=#1\p@
        \set@simple@size@args#2<#3\@nnil
        \execute@size@function\sizefn@info
        \ifx\external@font\@empty
        \else
          \let\reserved@f\@empty
        \fi
     \fi
     \reserved@f}
\def\set@size@funct@args{\@ifnextchar[%
  \set@size@funct@args@{\set@size@funct@args@[]}}
\def\set@size@funct@args@[#1]#2\@nil{%
  \def\mandatory@arg{#2}%
  \def\optional@arg{#1}}
%</2ekernel>
%<*2ekernel>
\def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
\@onlypreamble\DeclareSizeFunction
%</2ekernel>
```

```
%<*2ekernel|package>
\def\execute@size@function#1{%
%<*trace>
         \@ifundefined{s@fct@#1}%
           {\errmessage{Undefined font size function #1}%
            \s@fct@}%
           {\csname s@fct@#1\endcsname}%
%</trace>
%<-trace>      \csname s@fct@#1\endcsname
}
%</2ekernel|package>
%<*2ekernel>
\def\try@size@range{%
    \expandafter\extract@rangefontinfo\font@info ↊
          ↊<-*>\@nil<\@nnil
}
\gdef\try@size@substitution{%
  \@tempdimb \maxdimen
  \let \best@size \@empty
  \expandafter \try@simples \font@info <\number\@M>\@nil<\@nnil
}
\def\font@submax{0pt}
\def\fontsubfuzz{.4pt}
%</2ekernel>
%<+package>\def\fontsubfuzz{0pt}
%<*2ekernel>
\gdef\try@simples#1<#2>{%
  \tryif@simple#2->\tryif@simple}
\gdef\tryif@simple#1-#2\tryif@simple{%
  \let \reserved@f \try@simples
  \if>#2%
    \dimen@ #1\p@
    \ifdim \dimen@<\@M\p@
      \ifdim \f@size\p@<\dimen@
        \@tempdimc \dimen@
        \advance\@tempdimc -\f@size\p@
      \else
        \@tempdimc \f@size\p@
        \advance\@tempdimc -\dimen@
      \fi
      \ifdim \@tempdimc<\@tempdimb
        \@tempdimb \@tempdimc
        \def \best@size{#1}%
      \fi
    \else
% %\subst@size            %% coded inline
% %\def\subst@size{%
```

```
  \ifx \external@font\@empty
    \ifx \best@size\@empty
    \else
      \ifdim \@tempdimb>\font@submax \relax
        \xdef \font@submax {\the\@tempdimb}%
      \fi
      \let \f@user@size \f@size
      \let \f@size \best@size
      \ifdim \@tempdimb>\fontsubfuzz\relax
        \@font@warning{Font\space shape\space
            `\curr@fontshape'\space in\space size\space
             <\f@user@size>\space not\space available\MessageBreak
             size\space <\f@size>\space substituted}%
      \fi
      \try@simple@size
      \do@subst@correction
    \fi
 \fi
% %}
      \let \reserved@f \remove@to@nnil
    \fi
  \fi
  \reserved@f}
\DeclareSizeFunction{}{\empty@sfcnt\@font@warning}
\DeclareSizeFunction{s}{\empty@sfcnt\@font@info}
\def\empty@sfcnt#1{%
      \@tempdimb \f@size\p@
      \ifx\optional@arg\@empty
      \else
        \@tempdimb \optional@arg\@tempdimb
        #1{Font\space shape\space `\curr@fontshape'\space
           will\space be\MessageBreak
           scaled\space to\space size\space \the\@tempdimb}%
      \fi
      \edef\external@font{\mandatory@arg\space at\the\@tempdimb}}
\DeclareSizeFunction{gen}{\gen@sfcnt\@font@warning}
\DeclareSizeFunction{sgen}{\gen@sfcnt\@font@info}
\def\gen@sfcnt{%
      \edef\mandatory@arg{\mandatory@arg\f@size}%
      \empty@sfcnt}
\DeclareSizeFunction{genb}{\genb@sfcnt\@font@warning}
\DeclareSizeFunction{sgenb}{\genb@sfcnt\@font@info}
\def\genb@sfcnt{%
    \edef\mandatory@arg{\mandatory@arg\expandafter\genb@x\f@size..\@@}%
    \empty@sfcnt}
\def\genb@x#1.#2.#3\@@{\two@digits{#1}\genb@y#200\@@}
\def\genb@y#1#2#3\@@{#1#2}
```

```
\DeclareSizeFunction{sub}{\sub@sfcnt\@font@warning}
\DeclareSizeFunction{ssub}{\sub@sfcnt\@font@info}
\def\sub@sfcnt#1{%
    \edef\mandatory@arg{\f@encoding/\mandatory@arg}%
    \begingroup
     \expandafter\split@name\mandatory@arg/\@nil
     \try@load@fontshape
    \endgroup
    \let\f@user@size\f@size
```

Then we check whether this new combination is defined and give an error message if not. In this case we also switch to `\error@fontshape`.

```
    \expandafter
    \ifx\csname\mandatory@arg\endcsname\relax
      \errmessage{No\space declaration\space for\space
                  shape\space \mandatory@arg}%
      \error@fontshape
    \else
```

Otherwise we warn the user about the substitution taking place.

```
      #1{Font\space shape\space `\curr@fontshape'\space ↩
            ↪in\space
         size\space <\f@size>\space not\space ↩
              ↪available\MessageBreak
         Font\space shape\space `\mandatory@arg'\space ↩
              ↪tried\space
         instead}%
      \expandafter\split@name\mandatory@arg/\@nil
    \fi
```

Then we restart the font specification scan by calling `\get@external@font`.

```
    \edef\f@size{\f@user@size}%
    \get@external@font
    \do@subst@correction
}
```

The `subf` size function allows substitution of another font. The mandatory argument is the external name of the font to be substituted, the optional argument a size scaling factor like in the default size function. The main difference to the default size function is the warning message.

```
\DeclareSizeFunction{subf}{\subf@sfcnt\@font@warning}
\DeclareSizeFunction{ssubf}{\subf@sfcnt\@font@info}
\def\subf@sfcnt#1{%
      #1{Font\space shape\space `\curr@fontshape'\space ↩
            ↪in\space
         size\space \f@size\space not\space ↩
              ↪available\MessageBreak
        external\space font\space `\mandatory@arg'\space used}%
      \empty@sfcnt#1%
      }
```

The `fixed` size function is for using a font at a different size than requested. A warning message is printed, and the external font to be used is taken from the

mandatory argument. If an optional argument is present it is used as the 'at' size for the font. Otherwise the font is loaded at its design size.

```
\DeclareSizeFunction{fixed}{\fixed@sfcnt\@font@warning}
\DeclareSizeFunction{sfixed}{\fixed@sfcnt\@font@info}
```

```
\def\fixed@sfcnt#1{%
  \ifx\optional@arg\@empty
    \let\external@font\mandatory@arg
  \else
    \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
  \fi
  #1{External\space font\space `\external@font'\space loaded\space
     for\space size\MessageBreak
     <\f@size>}%
}
%</2ekernel>
```

phd package and give a background as to how fonts are used in LaTeX.

16.1 Introduction

Selecting the right fonts for a book is a job best left to the book designer. Despise this good advice most LaTeX authors get their hands dirty trying to play the role of the book designer. A word of advice is that most of them make a royal mess of it. Irrespective of the TeX engine employed, being TeX, LaTeX, LuaLaTeX or XeLaTeX there are two issues in using fonts. How to select them and specify them and what fonts to use. We will dwell on the technical aspects of font selection in this Chapter.

There is another more serious aspect in selecting fonts based on "physiological" considerations. Boris Veytsman in an article in TUGboat boris2012 reviewed the literature comparing fonts for readbility as well as the "trustability" of the text based on different fonts. Experiments carried out by Morris[1] concluded that fonts affect the reader's attitude towards the text. Baskerville scored the highest and Comic Sans the lowest. Interestingly Computer Modern, the default typeface of TeX, scored high in the test. Other tests carried out by **boris2012a** also concluded that there are no noticable differences between serif and non-serif fonts in reading comprehension for cyrillic adult readers and that comprehension and reading speed might be affected by factors other than the font serifs alone.

Of course the biggest effect on readers is when fonts are used for "branding" a product into consumers minds. Marketers have been brainwashing consumers for years through the use of fonts. In *Branding With Type* by Rogener, Pool, and Packhauser (1995), a fervent argument is made for unique but consistent typefaces as a crucial element of corporate branding. Rogener et al. describe the fonts used by IBM, Mercedes, Nivea, and Marlboro as instantly recognisable internationally, and imply that the significant investment by such companies in design and copyright of trademarked fonts is worthwhile.

For example, **rogener1995** discuss the Nivea Bold typeface developed in 1992 by Gunther Heinrich at advertising agency TBWA in Hamburg, Germany, for skincare brand Nivea, and claim that the Nivea Bold typeface has effectively embodied the Nivea brand's 'pure and simple' product philosophy. They link the font directly to profitability and Nivea's worldwide product category market share of 35%[2]

16.2 The Choice of Typesetting Engine

If you use only `pdfLaTeX` the range of fonts is rather limiting and I would highly recommend for any serious typesetting work to move onto `XeLaTeX` and the use of the package **fontspec** fontspec. Another alternative is to use LuaLaTeX. The latter is becoming more stable and is production ready to a large extend. It is expected to be the successor to pdfTeX.

One of the things I wanted to achieve with the **phd**

package was to take care of different TeX engines, and to ensure that the package can be used irrespective of the TeX engine used.

[1] **morris2012a**.
[2] **rogener1995**.

Before we start outlining the scheme let us start, by demonstrating how to load one of the standard fonts provided by LaTeX. We will load the Computer Modern font.

Example 2.1 How to load a font

```
\newcommand{\fontdemo}[4][OT1]{
    \leavevmode
    \textcolor{thefontname}{#2}
    \fontencoding{#1}\fontfamily{#3}\selectfont#4 }
\bgroup
\fontdemo{CM}{cmtt}{ \alphabet\par}

\fox
\egroup

\hrule
\vskip 1in
\centerline{\bf A SHORT STORY}
\vskip 6pt
\centerline{\sl    by A. U. Thor} % !`?`?! (modified)
\vskip .5cm
Once upon a time, in a distant
  galaxy called \"O\"o\c c,
there lived a computer
named R.~J. Drofnats.

Mr.~Drofnats---or ``R. J.,'' as
he preferred to be called---% error has been fixed!
was happiest when he was at work
typesetting beautiful documents.
\vskip 1in
\hrule
%\vfill\eject

\meaning\ttdefault
```

CM a b c d e f g h i j k l m n o p q r s t u v w x y z
The quick brown fox jumps over the lazy dogs.

A SHORT STORY

by A. U. Thor

Once upon a time, in a distant galaxy called Öö ç, there lived a computer named R. J. Drofnats.

Mr. Drofnats—or “R. J.,” as he preferred to be called—was happiest when he was at work typesetting beautiful documents.

macro:->SourceCodePro(0)

In the example we have used a number of convenience commands that are provided by the **phd** package.

`\alphabet`

Typesets the letters of the English alphabet

`\fox`

Typesets the fox passage

The example creates a convenience command to call the `computer modern typewriter` font and to print the alphabet.[3] In this case we are asking LaTeX to load a font from the `cmtt` family.

To load a font two things are required the encoding scheme [`OT1`] in the example and the somewhat cryptic font family name [`cmtt`].

16.3 What is a character? And a glyph?

A character is an abstract concept: the letter “A” is a character, while any particular drawing of that character is a glyph. In many cases there is one glyph for each character and one character for each glyph, but not always.

The glyph used for the Latin letter “A” may also be used for the Greek letter “Alpha”, while in Arabic writing most Arabic letters have at least four different glyphs (often vastly more) depending on what other letters are around them.

16.4 What's a font?

As the **fontinst**

manual says: “Once upon a time, this question was easily answered: a font is a set of type in one size, style, etc. There used to be no ambiguity, because a font was a collection of chunks of type metal kept in a drawer, one drawer for each font” fontinst.

With digital typesetting, things are more complicated. What a font *is* isn't easy to pin down. A typical use of a PostScript font with LaTeX might use these elements:

[3] The command `\alphabet` is provided by the phd package.

1. Type 1 printer font file
2. Bitmap screen font file
3. Adobe font metric file (afm file)
4. TEX font metric file (tfm file)
5. Virtual font file (vf file)
6. font definition file (fd file)

Looked at from a particular point of view, each of these files *is* the font. So what's going on? Every text font in LATEX has five attributes:

16.5 The low-level interface

The low level commands are mainly used to define new commands in packages and classes. It is helpful to understand the internal organization of fonts in LATEX's font selection scheme New Font Selection Scheme (NFSS). Although termed new, one needs to understand that this is now almost twenty years old.

One of the goals of LATEX's font selection scheme was to allow a rational selection with algorithms guided by the principles of generic markup companion.

Internally the LATEX kernel keeps track of five independent attributes. The "current encoding", the "current family", the "current series", the "current shape", and the "current size". This was introduced in NFSS release 2 after it became clear that real support in multiple languages would be possible only by maintaining the character-encoding scheme independly of the other font attributes.

16.5.1 Encoding Schemes

The *encoding* scheme (in the example `OT1`) provides information as to which glyph goes into what slot in a font table. These font tables can be printed using `fonttest.tex`. We show the test for `cmtt10` in Figure **??**. The most common values for the font encoding are:

OT1	TEX text
T1	TEX extended text
OML	TEX math italic
OMS	TEX math symbols
OMX	TEX math large symbols
U	Unknown
L⟨*xx*⟩	A local encoding

family The name for a collection of fonts, usually grouped under a common name by the font foundry. For example, 'Adobe Times', 'ITC Garamond',

and Knuth's 'Computer Modern Roman' are all font families.

There are far too many font families to list them all, but some common ones are:

cmr	Computer Modern Roman
cmss	Computer Modern Sans
cmtt	Computer Modern Typewriter
cmm	Computer Modern Math Italic
cmsy	Computer Modern Math Symbols
cmex	Computer Modern Math Extensions
ptm	Adobe Times
phv	Adobe Helvetica
pcr	Adobe Courier

series How heavy or expanded a font is. For example, 'medium weight', 'narrow' and 'bold extended' are all series.

shape The form of the letters within a font family. For example, 'italic', 'oblique' and 'upright' (sometimes called 'roman') are all font shapes. The most common values for the font shape are:

n	Normal (that is 'upright' or 'roman')
it	Italic
sl	Slanted (or 'oblique')
sc	Caps and small caps
ui	upright italic shape
ol	ouline shape

To change the shape attribute the fontshape is used to change the shape attribute. If you try this and it does not work, it means that the font you have selected does not have the font shape you have requested. You may need to specify an appropriate font family as well:

```
Example 5.2 usefont

{\usefont{T1}{cmr}{m}{sc}  \raggedright \lorem}
```

FUSCE ADIPISCING JUSTO NEC ANTE. NULLAM IN ENIM. PELLENTESQUE FELIS ORCI, SAGITTIS AC, MALESUADA ET, FACILISIS IN, LIGULA. NUNC NON MAGNA SIT AMET MI ALIQUAM DICTUM. IN MI. CURABITUR SOLLICITUDIN JUSTO SED QUAM ET QUADD.PAR

size The design size of the font, for example '10pt'. If no dimension is specified, 'pt' is assumed.

These five parameters specify every LaTeX font, for example:

LaTeX specification	Font	TEX font name

`OT1 cmr m n 10`	Computer Modern Roman 10 point	cmr10
`OT1 cmss m sl 1pc`	Computer Modern Sans Oblique 1 pica	cmssi12
`OML cmm m it 10pt`	Computer Modern Math Italic 10 point	cmmi10
`T1 ptm b it 1in`	Adobe Times Bold Italic 1 inch	ptmb8t at 1in

Example 5.3 Th usefont command

```
\bgroup
\fontsize{12}{14pt} \lorem
\lorem
\egroup
```

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par
Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

When you get a font error or an underfull or overfull box TEX always will print an error with the font specification in full as shown below:

```
LaTeX Font Warning: Font shape `EU1/cmr/m/sc' undefined
(Font)              using `EU1/cmr/m/n' instead on input line 160.
```

16.6 Setting several font attributes

Often it is required that several attributes of a particular font need to be set. For this task LATEX provides the command usefont. This command takes four arguments: the encoding, family, series, and shape. The command updates those and then calls selectfont. If you also need to change the size and baseline skip, place a fontsize command in front of it.

Example 6.4 Th usefont command

```
\bgroup
\fontsize{12}{14pt}\usefont{OT1}{cmdh}{bc}{it}
\lorem
\egroup
```

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

Do note that when you use the **hyperref**

, you will get a surprise, all the commands have been converted to ”PU” encoding. This is mostly harmless and is done in order for **hyperref** to mark bookmarks[4] in a safe way.

Example 6.5 font encoding

```
\meaning\textasciitilde\\
\meaning\"\\
\meaning\NG\\
\meaning\k\\
\meaning\alpha
\meaning\printfontparams

\printfontparams
```

macro:->\TU-cmd \textasciitilde \TU\textasciitilde
macro:->\TU-cmd \”\TU\”
macro:->\TU-cmd \NG \TU\NG
macro:->\TU-cmd \k \TU\k
\mathchar”010Bundefined

A peek at the **puenc.def** reveals the inner workings of the encoding mechanism.

```
\ProvidesFile{puenc.def}
  [2003/01/20 v6.73l
  Hyperref: PDF Unicode definition (HO)]
\DeclareFontEncoding{PU}{}{}
\DeclareTextCommand{\textLF}{PU}{\80\012} % line feed
\DeclareTextCommand{\textCR}{PU}{\80\015} % carriage return
\DeclareTextCommand{\textHT}{PU}{\80\011} % horizontal tab
\DeclareTextCommand{\textBS}{PU}{\80\010} % backspace
\DeclareTextCommand{\textFF}{PU}{\80\014} % formfeed
\DeclareTextAccent{\`}{PU}{\textgrave}
\DeclareTextAccent{\'}{PU}{\textacute}
\DeclareTextAccent{\^}{PU}{\textcircumflex}
```

LaTeX uses a number of other files to get to the particular file that contains the font metrics file **cmtt10** and to find the appropriate file. For the original Knuth fonts the filenames have been kept the same, essentially as a request from Knuth that one should not change them.

Most of the difficulty in selecting and using fonts is figuring the encoding scheme and the Karl Berry naming scheme. In the Example **??** we select the **\fontfamily cmtt** which is computer modern type writer and then we invoke the macros for the shape **\itshape** and print the **alphabet**. The macro **\alphabet** is build-in the **phd** package as we use it in a few places.

[4] http://tex.stackexchange.com/questions/198810/why-does-the-hyperref-package-changes-encoding-of-font-commands

Test of cmtt10 on August 23, 2014 at 0607

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	˝0x
´01x	Φ	Ψ	Ω	↑	↓	'	¡	¿	
´02x	ı	j	`	´	˘	˜	¯	˙	˝1x
´03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
´04x	␣	!	"	#	$	%	&	’	˝2x
´05x	(	)	*	+	,	-	.	/	
´06x	0	1	2	3	4	5	6	7	˝3x
´07x	8	9	:	;	<	=	>	?	
´10x	@	A	B	C	D	E	F	G	˝4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	˝5x
´13x	X	Y	Z	[	\	]	^	_	
´14x	‘	a	b	c	d	e	f	g	˝6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	˝7x
´17x	x	y	z	{	|	}	~	¨	
	˝8	˝9	˝A	˝B	˝C	˝D	˝E	˝F	

Figure 16.1: Output from testfont.tex for cmtt10 font

16.6.1 The Postscript fonts

With Adobe reader a number of fonts come pre-packaged and these have been incorporated into LaTeX 2e. These fonts can be found in all TeX distributions. The *Times New Roman* is named **ptm**.

Example 6.6 The Postscript fonts

```
\raggedright
\begin{tabular}{@{}>{\sffamily\bfseries}rl}
\fonttitle{The Adobe `LaserWriter 35', 10 typefaces in a total of 35
different styles, standard on all PostScript printers}

%\thefont{Avant Garde Book}{pag}{\fontsize{9}{9}\selectfont\sample}
%\thefont{Bookman Light}{pbk}{\sample}
\thefont{Courier}{pcr}{\sample}
\thefont{Helvetica}{phv}{\sample}
\thefont{New Century Schoolbook}{pnc}{\sample}
\thefont{Palatino}{ppl}{\sample}
\thefont[U]{Symbol}{psy}{\sample}
\thefont{Zapf Chancery Medium Italic}{pzc}{\fontsize{12}{12}\selectfont\itshape\sample}
\thefont[U]{Zapf Dingbats}{pzd}{\sample}
\end{tabular}
```

The Adobe 'LaserWriter 35', 10 typefaces in a total of 35 different styles, standard on all PostScript printers

Courier	Typographia Ars Artium Omnium
Helvetica	Typographia Ars Artium Omnium
New Century Schoolbook	Typographia Ars Artium Omnium
Palatino	Typographia Ars Artium Omnium
Symbol	Τψπογραπηια Αρσ Αρτιυμ Ομνιυμ
Zapf Chancery Medium Italic	*Typographia Ars Artium Omnium*
Zapf Dingbats	✳❙❒❑✻❒❁❒✻✻✻❁ ✡❒▲ ✡❒▼✻◆❍ ★❍■✻◆❍

Using the **phd** package we can come closer to the **fontspec** or LuaTeX way of doing things and use longer font names as those found in the operating system.

Activating the key will set the font to **pzc** and unless is within a group will typeset the rest of the document with this typeface.

`/key/font name` = {⟨*Zapf Chancery Medium Italic*⟩}

This is how it is typeset

`/phd/font name` = {⟨*Old Standard*⟩}

Setting the key to `Old Standard` will typeset the next sample in `OldStandard-Regular`, **`Stylistic Set=2`**.

Alice had never been in a court of justice before, but she had read about them in books, and she was quite pleased to find that she knew the name of nearly everything there. "That's the judge," she said to herself, "because of his great wig.".

The judge, by the way, was the King, and as he wore his crown over the wig, (look at the frontispiece if you want to see how he did it,) he did not look at all comfortable, and it was certainly not becoming.

abcdefg

`/phd/font name = {`⟨*Junicode*⟩`}`

Setting the key to `Junicode` will typeset the next sample in `Junicode`, `Stylistic Set=2`.

Alice had nevep been in a coupt op jupticе bepope, but phe had pead about them in bookp, and phe pap quite pleaped to pind that phe knep the name op neaply evepypthing thepe. ``That'p the judge,'' phe paid to hepphelp, ``becaupe op hip gpeat pig.''.

The judge, by the pay, pap the King, and ap he pope hip cpopn ovep the pig, (look at the ppontippiece ip you pant to pee hop he did it,) he did not look at all compopptable, and it pap cepptainly not becoming.

abcdepg

`/phd/font name ={`⟨*Bookman Light or bookman*⟩`}`

Bookman Light or **`bookman`**

"And that's the jury-box," thought Alice, "and those twelve creatures," (she was obliged to say "creatures," you see, because some of them were animals, and some were birds) "I suppose they are the jurors.". She said this last word two or three times over to herself being rather proud of it: for she thought, and rightly too, that very few little girls of her age knew the meaning of it at all. However, "jurymen" would have done just as well.

`/phd/font name  ={`⟨*Utopia or utopia*⟩`}`

goats are animals found in all sort of places. The paragraph has been set using the font family **`utopia`**. The comment about the goats was just to get the letter g. comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice

goats are animals found in all sort of places. The paragraph has been set using the font family **`utopia`**. The comment about the goats was just to get the letter g. comfortable in mountain areas. I don't recall Alice They

are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice They are more comfortable in mountain areas. I don't recall Alice

Goats are among the earliest animals domesticated by humans. The most recent genetic analysis confirms the archaeological evidence that the wild Bezoar ibex of the Zagros Mountains are the likely origin of almost all domestic goats today. Neolithic farmers began to herd wild goats for easy access to milk and meat, primarily, as well as for their dung, which was used as fuel, and their bones, hair, and sinew for clothing, building, and tools. The earliest remnants of domesticated goats dating 10,000 years before present are found in Ganj Dareh in Iran. Goat remains have been found at archaeological sites in Jericho, Choga Mami Djeitun and Çayönü, dating the domestication of goats in Western Asia at between 8000 and 9000 years ago.[5]

As you have observed we did not change the normal size of paragraphs, but the examples demonstrate that differences in font families also affect the visual size of the typeset text. `Helvetica` is normally scaled down to 0.95 and `Chancery` is scaled a little bit up or we use a larger font size.

16.6.2 Additional free fonts for use with LaTeX

A number of archaic and other fonts are available in the LaTeX historical collection. These are very impressive. They also provide in most instances transliterations.

The Historical Collection

Cypriot 𐠋𐠊𐠌𐠈𐠎𐠅𐠄𐠏𐠐𐠉 𐠀𐠄𐠊 𐠀𐠄𐠇𐠐𐠊𐠉𐠎 𐠋𐠎𐠅𐠐𐠊𐠉𐠎

Linear 'B' Typographia Ars Artium Omnium

Phoenician 𐤂𐤉𐤋𐤏𐤓𐤊𐤋𐤇𐤔 𐤁𐤦 𐤁𐤉𐤓𐤕 𐤌𐤋𐤌

Runic ᛏᛃᛈᚩᚷᚱᛗᛈᚺᛁᛗ ᛗᚱᛋ ᛗᚱᛏᛁᚢᛗ ᚩᛗᛏᛁᚢᛗ ᚩᛏᛋᛗᚱᛗᛏᚱᛁᛉ

Bard 𝔗𝔶𝔭𝔬𝔤𝔯𝔞𝔭𝔥𝔦𝔞 𝔄𝔯𝔰 𝔄𝔯𝔱𝔦𝔲𝔪 𝔒𝔪𝔫𝔦𝔲𝔪

Uncial Typographia Ars Artium Omnium

16.6.3 Uncial fonts

Peter Wilson's **uncial**

package provides a useful uncial font and is easily used by just including the file.

[5]Text is from wikipedia's article for the domesticated goat.

Example 6.7 Unical fonts example

```
\begin{center}
The Uncial Huge normal font. \\ \par
{\unclfamily\Huge \ABC\\ \alphabet\\ \punct{}\dashes\\
    \figs\\ \par }
\end{center}
```

The Uncial Huge normal font.

16.7 Dingbat and Symbol Fonts

Fonts containing collections of special symbols, which are normally not found in a text font, are called *dingbats*. One such font, the PostScript font Zapf Dingbats, is available if you use the **pifont** package, originally written by Sebastian Rahtz, and now part of **PSNFSS**. This is loaded automatically by the **phd** package. (See also implementation code at Page **??**).

The parameter for the **\ding** command is an integer that specifies the character to be typeset according to Table **??**. For example **\ding{38}** gives ✆.

For Open Type fonts the **Wingdings** family can be found on Windows systems. The advent of Unicode and the universal character set allowed commonly used dingbats to be given their own character codes. Although fonts claiming Unicode coverage will contain glyphs for dingbats *in addition* to alphabetic characters continue to be popular, primarily for ease of input. Such fonts are sometimes known as *pi fonts*.

16.7.1 Unicode Dingbats block

The Dingbats block **U+2700–U+27BF** was added to the Unicode Standard in June, 1993, with the release of version 1.1. This code block contains decorative character variants, and other marks of emphasis and non-textual symbolism. Most of its characters were taken from Zapf Dingbats.

The Ornamental Dingbats block (**U+1F650–U+1F67F**) was added to the Unicode Standard in June 2014 with the release of version 7.0. This code block contains ornamental leaves, punctuation, and ampersands, quilt squares, and checkerboard patterns. It is a subset of dingbat fonts Webdings, Wingdings, and Wingdings 2. [6]

A font that we will be using for many of the XƎLaTeXexamples is **code2000** and **code2001**. The fonts were designed by James Kas [7]. They are True type fonts. The fonts contain a respectable collection of more or less exotic Unicode characters both within the Basic Multilingual Plane (BMP). They were designed by James Kass and were freeware. Sadly the website is no longer available, but the files can be downloaded in the links I have provided. I have also included them in the distribution for the **phd** package, as they are such a useful tool.

\codetwothousand

Loads the TrueType font `code2000.ttf`

\codetwothousandone

Loads the TrueType font `code2001`

\symbola

Loads the TrueType font `symbola`

```
\newfontfamily{\codetwothousand}{code2000.ttf}
  \newfontfamily{\codetwothousandone}{code2001.ttf}
  \newfontfamily{\symbola}{symbola.ttf}
```

[6] See http://std.dkuug.dk/jtc1/sc2/wg2/docs/n4115.pdf
[7] They can be downloaded at http://www.alanwood.net/downloads/index.html

Example 7.8 Wingdings

```
\ifxetex
   {\codetwothousand \symbol{9742} \symbol{9743}
    Katakana (, )
   \codetwothousandone \symbol{57508}
   \symbola \symbol{9816}
  }
\else
  \ifluatex
  {\codetwothousand \symbol{9742} \symbol{9743}
    Katakana (, )
   \codetwothousandone \symbol{57508}
   \symbola \symbol{9816}
  }
  \else
   Compile the document with XeTeX to see the example
  \fi
\fi
```

Katakana (,) 🤚

Another useful font for experimenting if you are using a Windows computer is `Arial Unicode MS`.

The `Symbola Font` has many other symbols, including chess and even Mahjong symbols.[8]. George Douros has packaged many of the fonts for archaic languages, but sadly the substitution mechanisms of LaTeX do not always map the fonts properly.

With `LuaLaTeX` and `XeLaTeX`, TeX has moved into the twenty-first century and its usefulness can now be extended to many other languages and fields.

16.8 Naming digital fonts

Commercial and Open Source fonts come as a set of several files. The `.pfb` file and less frequently, a `.pfa` file or other files depending on the type of font and the operating system and provider. The metric information file resides in an associated `.afm` file. Other files, with extensions `.inf` (information) and `.pfm` are irrelevant to LaTeX and TeX.

Fonts already have names given them by their designers. The problem lies in associating this name with the font files. Restriction of operating systems originally from PC-DOS dates, restricted to the initial part of file names to eight characters.

[8] http://users.teilar.gr/~g1951d/Symbola.pdf

16.8.1 Karl Berry naming scheme

The original inspiration for Fontname was Frank Mittelbach and Rainer Schoepf's article in TUGboat 11(2) (June 1990). This led to an article by Karl Berry in TUGboat 11(4) (November pages 512-519).

Karl Berry then suggested a system—with many limitations, but perhaps the best that could have been done in its time, for mapping a lengthy font name into a file name that was eight or fewer characters long. If the font files are renamed accordingly then we can deduce the nature of the font by examining its file name. The scheme did not apply to the original Computer Modern fonts that retained their original names fontname.

This scheme assumes that only eight characters or fewer can be available for naming the font. These eight characters look like,

```
FNNW[S][V]7V
```

Some additional comments on this shorthand notation is in order. The most common foundry abbreviations are **p** for Adobe (from PostScript), **b** for BitStream, and **m** Monotype. A font flouting this scheme will begin with a z.

The next two letters are reserved for the typeface name. The hundreds and hundred of available faces guarantees that many of these will be cryptic, even for the most common typefaces—Adobe Garamond is **ad**.

16.9 The phd package interface.

By design feature options for XeTeX/XeLaTeX have currently been restricted. The reason behind this decision is that I was concerned that I would have added a complicated interface with very little reason as to its use. I opted for a more semantic approach and expect the user to define custom macros to handle anything else.

`mainfont` = {⟨*font1,font2,font3*⟩} A comma separated list of one or more font-names. The main font will be set to the first font found.

`chapterfont` = {⟨*font1,font2,font3*⟩} A comma separated list of one or more font-names. The main font will be set to the first font found.

`sectionfont` = {⟨*font1,font2,font3*⟩} A comma separated list of one or more font-names. The main font will be set to the first font found.

`contentsfont` = {⟨*font1,font2,font3*⟩} A comma separated list of one or more font-names. The main font will be set to the first font found.

`/phd/bibliography font` = {⟨*font1,font2,font3*⟩}
A comma separated list of one or more font-names. The main font will be set to the first font found.

Note that the package will first check if is running under XeTeX. If it does it will execute the commands and load the macros, otherwise it will fall back on pdfLaTeX commands.

16.10 Viewing and selecting fonts

16.10.1 Typefaces that come with the standard LaTeX distribution

Computer Modern (CM), LaTeX's default typeface

CM Roman	Typographia Ars Artium Omnium
CM Italic	*Typographia Ars Artium Omnium*
CM Slanted (Oblique)	*The quick brown fox jumps over the lazy dogs.*
CM Bold	**Typographia Ars Artium Omnium**
CM Bold Extended	**Typographia Ars Artium Omnium**
CM Bold Italic	***Typographia Ars Artium Omnium***
CM Bold Slanted	***Typographia Ars Artium Omnium***
CM Caps & Small Caps	Typographia Ars Artium Omnium
CM Sans-Serif	Typographia Ars Artium Omnium
CM Sans-Serif Oblique	*Typographia Ars Artium Omnium*
CM Sans-Serif Bold	**Typographia Ars Artium Omnium**
CM Typewriter	`Typographia Ars Artium Omnium`
CM Typewriter Italic	*`Typographia Ars Artium Omnium`*
CM Typewriter Bold	`Typographia Ars Artium Omnium`
CM Typewriter C&SC	`Typographia Ars Artium Omnium`
CM Mathematics	$E = mc^2$
CM 'Dunhill'	Typographia Ars Artium Omnium
CM 'Fibonacci'	Typographia Ars Artium Omnium

16.11 Discussion

Unfortunately, even with the best will loading fonts will always be a difficult task in TeXHopefully the interface provided will result in better separation of presentation from content and offers consistency in the styling of documents. Nothing prevents you from adding normal macros to styles. Each style can be treated as a package in many respects.

16.12 XeLaTeX and LuaLaTeX

Verdana

The quick brown fox jumps over the lazy dogs.
a b c d e f g h i j k l m n o p q r s t u v w x y z
+ ? / |

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on a high and caught it, and this ball was her favorite plaything.

Calibri

The quick brown fox jumps over the lazy dogs.
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
+ ? / |

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on a high and caught it, and this ball was her favorite plaything.
Θαμκυαμ πλαθονεμ ραθιονιβυς ναμ ει, δυις περπετυα σιθ αδ, νες ιδ δισυντ σοντεντιωνες. Κυι σινθ μυνδι εα, φιμ αν γραεσω ιυδισαβιτ, εραθ δολορ φιρθυθε υθ δυο. Συ νοσθερ οπθιων ευμ, μει ερος προβο φιερενθ ευ. Ιυς μανδαμυς τωρκυαθος εξπεθενδις ιδ. Σεδ θε νιβχ νονυμυ δελισαθισιμι, φιμ νο νιβχ λαβωραμυς, σεα εα δισο ποσιμ αντιωπαμ.

16.13 Utilities for testing fonts

The package **fonttable** is an extension and re-implementation of Donald Knuth's `testfont.tex`, which is available from CTAN. The package was developed by Peter Wilson and currently maintained by Will Robertsonfonttable. It provides a number of utility commands for typesetting font tables.

The fonttable{⟨*font*⟩} takes the font file as an argument and typesets it in a nice table.

A great tool to inspect a True Type font on the command line is `luaotfload-tool`:

```
luaotfload-tool --find="Iwona" --inspect
```

16.14 `.tfm` files

When you tell TEX that you will be using a particular font, it has to find out information about that font. This information is stored in a file with the extension `.tfm`. For example when you say:

```
\font a=cmr10
```

TEX looks for a file named **`cmr10.tfm`**. If this is not found then an error is issued **`Lookup failed on file CMR10.TFM`**

Generall speaking, a font's **`.tfm`** file contains information about the height, width and depth of all the characters in the font plus kerning and ligature information. So, **`cmr10.tfm`** might say that the lower-case "d" in CMR10 is 5.5 points wide, 6.94 points high, etc. This is the information that TEX uses to make its lowest-level boxes—those around characters. See the TEX manual for information about what TEX does with these boxes. Note the **`.tfm`** files do not contain any information that is device dependent. Only device-drivers read TEX's **`dvi`** output files can use that sort of information.

16.15 Fonts for Far East Languages

In internationalization, CJK is a collective term for the Chinese, Japanese, and Korean languages, all of which use Chinese characters and derivatives (collectively, CJK characters) in their writing systems. Occasionally, Vietnamese is included, making the abbreviation CJKV, since Vietnamese historically used Chinese characters as well. The characters are known as hànzì in Chinese, kanji in Japanese, hanja in Korean, and Ch Nôm in Vietnamese.

16.15.1 Selecting a font

The easiest way is with Will Robertson's **fontspec**

package. In this sample, we have used the `SimSun` font, which can be found on windows machines:

```
\usepackage{fontspec}
\setromanfont{SimSun}
```

in the preamble, to use a Far Eastern font as the initial default typeface.

16.16 Entering CJK text

You can just enter Unicode text directly in the document: . Don't use legacy LaTeX packages such as `inputenc` or `CJK`, as XeTeX reads the text as Unicode characters, not the separate byte codes of UTF-8 sequences, and passes them directly to the Unicode font. (Actually, it would probably be possible to use `\XeTeXinputencoding "bytes"` and work with legacy LaTeX input encoding support. But then you're pretty much committed to all the old encoding and font machinery, and there's not much point in using the XeTeX engine at all.)

16.17 Changing the font size in LaTeX

Changing the font size in LaTeX can be done at two levels, affecting the whole document or elements with in it. Using a different font size on a global level will affect all normal-sized text as well as the sizes of headings, footnotes, etc. By changing the font size locally, however, a single word, a few lines of text, a large table, or a heading throughout the document may be modified. Fortunately, there is no need for the writer to juggle with numbers when doing so. LaTeX provides a set of macros for changing the font size locally, taking into consideration the document's global font size. thurnherr2012

16.17.1 Changing the font size on the document-wide level

The standard classes article, report and book support three different font sizes: 10pt, 11pt, 12pt. By default, the font size is set to 10pt and can be modified by passing any of the previously-mentioned value as a class option. As an example, suppose you want to change the font size for normal text to 12pt throughout the document. For the class report, this is how you would do that:

```
\documentclass[12pt]{report}
```

In most cases, the available font sizes for the standard classes are sufficient and you do not have to bother about loading special packages that provide more options.

16.18 Changing the font size locally

A common scenario is that the author of a document needs to change the font size for a word or paragraph, decrease the font size of a large table to make it fit on a page or increase the size of a heading throughout the document. LaTeX implements a set of macros which allow changing font size from Huge to tiny, literally. That way, the author does not have to worry about numbers. The macros, including the exact font size in points, are summarized in table 1. A good rule of thumb is not to use too many different sizes and not to make things too small or too big.

LaTeX provides two different ways to use these font size modifier macros: inline or as an environment using `\begin...\end`:

Example 18.9 Changing the size locally

```
{\Large This is some large text.\par}
\begin{footnotesize}
This is some footnote-sized text.
\end{footnotesize}
```

This is some large text.

This is some footnote-sized text.

The `\par` command at the end of the inline example adjusts baselineskip, the minimum space between the bottom of two successive lines

A number of packages exist **moresize**[9] moresize, **anyfont**[10] by **anyfont**. The standard classes, **memoir KOMA** classes and most journals also provide their own defined sizing commands.

Class option	10pt	11pt	12pt
`\Huge`	25pt	25pt	25pt
`\huge`	20pt	20pt	25pt
`\LARGE`	17pt	17pt	20pt
`\Large`	14pt	14pt	17pt
`\large`	12pt	12pt	14pt
`\normalsize` (default)	10pt	11pt	12pt
`\small`	9pt	10pt	11pt
`\footnotesize`	8pt	9pt	10pt
`\scriptsize`	7pt	8pt	8pt
`\tiny`	5pt	6pt	6pt

[9] http://www.ctan.org/pkg/moresize
[10] http://www.ctan.org/pkg/anyfont

phd package and give a background as to how fonts are used in LaTeX.

This file contains the top level LaTeX interface to the font selection scheme commands. See other parts of the LaTeX distribution, or The LaTeX Companion *for higher level documentation of these commands.*

17.1 NFSS Initialisation

```
%<*2ekernel>
```

17.1.1 Providing math *versions*

LaTeX provides two *versions*. We call them normal and bold, respectively.

```
\DeclareMathVersion{normal}
\DeclareMathVersion{bold}
```

```
\DeclareRobustCommand\rmfamily
        {\not@math@alphabet\rmfamily\mathrm
         \fontfamily\rmdefault\selectfont}
\DeclareRobustCommand\sffamily
        {\not@math@alphabet\sffamily\mathsf
         \fontfamily\sfdefault\selectfont}
\DeclareRobustCommand\ttfamily
        {\not@math@alphabet\ttfamily\mathtt
         \fontfamily\ttdefault\selectfont}
```

```
\DeclareRobustCommand\bfseries
        {\not@math@alphabet\bfseries\mathbf
         \fontseries\bfdefault\selectfont}
\DeclareRobustCommand\mdseries
        {\not@math@alphabet\mdseries\relax
         \fontseries\mddefault\selectfont}
\DeclareRobustCommand\upshape
        {\not@math@alphabet\upshape\relax
         \fontshape\updefault\selectfont}
```

```
\DeclareRobustCommand\slshape
        {\not@math@alphabet\slshape\relax
         \fontshape\sldefault\selectfont}
\DeclareRobustCommand\scshape
        {\not@math@alphabet\scshape\relax
         \fontshape\scdefault\selectfont}
\DeclareRobustCommand\itshape
        {\not@math@alphabet\itshape\mathit
         \fontshape\itdefault\selectfont}
```

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\eminnershape}{\eminnershape}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\em
        {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
                        \eminnershape \else \itshape \fi}%
```

```
\def\eminnershape{\upshape}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
```

```
%<latexrelease>\IncludeInRelease{0000/00/00}{\eminnershape}{\eminnershape}%
%<latexrelease>\DeclareRobustCommand\em
%<latexrelease>        {\@nomath\em \ifdim ↊\fontdimen\@ne\font >\z@
%<latexrelease>                        \upshape \else ↊\itshape \fi}%
%<latexrelease>\let\eminnershape\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
\def\not@math@alphabet#1#2{%
   \relax
   \ifmmode
     \@latex@error{Command \noexpand#1invalid in math mode}%
        {%
         Please
         \ifx#2\relax
            define a new math alphabet^^J%
            if you want to use a special font in math mode%
          \else
            use the math alphabet \noexpand#2instead of
            the #1command%
         \fi
         .
        }%
   \fi}
\def\boldmath{\@nomath\boldmath
              \mathversion{bold}}
\def\unboldmath{\@nomath\unboldmath
              \mathversion{normal}}
\def\math@version{normal}
\def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
\def\symbol#1{\char #1\relax}
\def\@setfontsize#1#2#3{\@nomath#1%
    \ifx\protect\@typeset@protect
      \let\@currsize#1%
    \fi
    \fontsize{#2}{#3}\selectfont}
%<*compat>
\def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
%</compat>
\def\oldstylenums#1{%
   \begingroup
    \spaceskip\fontdimen\tw@\font
    \usefont{OML}{\rmdefault}{\f@series}{it}%
    \mathgroup\symletters #1%
   \endgroup
}
```

To set up LaTeX's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple **\ifcase**.

```
\def\hexnumber@#1{\ifcase\number#1
 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
 9\or A\or B\or C\or D\or E\or F\fi}
```

\nfss@text

In it simplest form **\nfss@text** is an **\mbox**. This will produce unbreakable text outside math and inside math you will get text with the same fonts as outside. The only drawback is that such item won't change sizes in subscripts. But this behavior can be easily changed. With the `amstex` style option one will get a sub style called `amstext` which will redefine the **\nfss@text** macro to produce correct text in all sizes.

We have to use **\def** instead of the shorter **\let** since **\mbox** is undefined when we reach this point.

```
\def\nfss@text#1{{\mbox{#1}}}
```

(*End 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.

```
%\DeclareRobustCommand\copyright
%    {{\ooalign{\hfil
%     \raise.07ex\hbox{\mdseries\upshape c}\hfil\crcr
%     \mathhexbox20D}}}
```

\normalfont

The macro **\reset@font** is used in LaTeX to switch to a standard font, in order to initialize the current font in situations where typesetting is done in a new visual context (e.g. in a footnote). We define it here to allow the test for the new LaTeX version above but nevertheless are able to run all kind of mixtures. The user interface name for **\reset@font** is **\normalfont**: Added **\relax** after **\usefont**, as the latter eats up spaces.

```
\DeclareRobustCommand\normalfont
                  {\usefont\encodingdefault
                           \familydefault
                           \seriesdefault
                           \shapedefault
                   \relax}
\let\reset@font\normalfont
```

We left out the special LaTeX fonts which are not automatically included in the base version of the font selection since these fonts contain only a few characters which are also included in the AMS fonts so anybody who is using these fonts doesn't need them. But for compatibility reasons we will define these symbols.

```
\def\not@base#1{\@latex@error
  {Command \noexpand#1not provided in base LaTeX2e}%
```

```
  {Load the latexsym or the amsfonts package to
   define this symbol}}
\def\mho{\not@base\mho}
\def\Join{\not@base\Join}
\def\Box{\not@base\Box}
\def\Diamond{\not@base\Diamond}
\def\leadsto{\not@base\leadsto}
\def\sqsubset{\not@base\sqsubset}
\def\sqsupset{\not@base\sqsupset}
\def\lhd{\not@base\lhd}
\def\unlhd{\not@base\unlhd}
\def\rhd{\not@base\rhd}
\def\unrhd{\not@base\unrhd}
```

We now initialize all variables set by `\DeclareErrorFont`. These values are not really important since they will be overwritten later on by the definition in `fontdef.ltx`.

However, if `fontdef.cfg` is corrupted then at least a hopefully suitable error font is present.

```
\DeclareErrorFont{OT1}{cmr}{m}{n}{10}  %% don't modify this setting
                                       %% overwrite it in fontdef.cfg
                                       %% if necessary
\ifnum\inputlineno=\m@ne
\InputIfFileExists{fonttext.cfg}
           {\typeout{===================================^^J%
                     ^^J%
                      Local config file fonttext.cfg used^^J%
                     ^^J%
                     ===================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\typeout{!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
                     !^^J%
                     ! You MUST use a fonttext.cfg file!^^J%
                     ! As you are still using TeX2!!!!!^^J%
                     !^^J%
                     ! See the documentation file tex2.txt^^J%
                     !^^J%
                     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!}%
                    \batchmode \@@end}
\else
\InputIfFileExists{fonttext.cfg}
           {\typeout{===================================^^J%
                     ^^J%
                      Local config file fonttext.cfg used^^J%
                     ^^J%
                     ===================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{fonttext.ltx}}
```

```
\fi
\let\@addtofilelist\@gobble
```

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

```
\InputIfFileExists{fontmath.cfg}
           {\typeout{=====================================^^J%
                     ^^J%
                      Local config file fontmath.cfg used^^J%
                     ^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{fontmath.ltx}}
\let\@addtofilelist\@gobble
```

Then we preload several fonts. This file might be customized *without* changing the behavior of the format (i.e. necessary font definitions will be loaded at runtime if they are not preloaded). This is done in the file `preload.ltx`.

```
\InputIfFileExists{preload.cfg}
   {\typeout{=====================================^^J%
             ^^J%
             Local config file preload.cfg used^^J%
             ^^J%
             ====================================}%
     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
   }
   {\input{preload.ltx}}
\let\@addtofilelist\@gobble
```

```
\let\@acci\' \let\@accii\` \let\@acciii\=
```

```
%</2ekernel>
```

phd package and give a background as to how fonts are used in LaTeX.

18.1 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.

18.2 The `fonttext.ltx` file

```
%<*text>
\typeout{=== Don't modify this file, use a .cfg file instead ↊
      ↊===^^J}
\input {omlenc.def}
\input  {t1enc.def}
\input {ot1enc.def}       % <- should come after T1 for speed
\input {omsenc.def}
\fontencoding{OT1}
\DeclareFontEncodingDefaults{}{}
\DeclareFontSubstitution{T1}{cmr}{m}{n}
\DeclareFontSubstitution{OT1}{cmr}{m}{n}
\begingroup
\nfss@catcodes
\input  {t1cmr.fd}
\input  {ot1cmr.fd}
\endgroup
\begingroup
\nfss@catcodes
\input {ot1cmss.fd}
\input {ot1cmtt.fd}
\endgroupfontdef
\DeclareErrorFont{OT1}{cmr}{m}{n}{10}
\newcommand\rmdefault{cmr}
\newcommand\sfdefault{cmss}
\newcommand\ttdefault{cmtt}
\newcommand\bfdefault{bx}
\newcommand\mddefault{m}
```

```
\newcommand\itdefault{it}
\newcommand\sldefault{sl}
\newcommand\scdefault{sc}
\newcommand\updefault{n}
\newcommand\encodingdefault{OT1}
\newcommand\familydefault{\rmdefault}
\newcommand\seriesdefault{\mddefault}
\newcommand\shapedefault{\updefault}
%</text>
%<*math>
\typeout{=== Don't modify this file, use a .cfg file instead ===^^J}
\DeclareFontEncoding{OML}{}{}
\DeclareFontEncoding{OMS}{}{}
\DeclareFontEncoding{OMX}{}{}
\DeclareFontEncoding{U}{}{\noaccents@}
\DeclareFontSubstitution{OML}{cmm}{m}{it}
\DeclareFontSubstitution{OMS}{cmsy}{m}{n}
\DeclareFontSubstitution{OMX}{cmex}{m}{n}
\DeclareFontSubstitution{U}{cmr}{m}{n}
\begingroup
\nfss@catcodes
\input  {omlcmm.fd}
\input  {omscmsy.fd}
\input  {omxcmex.fd}
\input  {ucmr.fd}
\endgroup
\DeclareSymbolFont{operators}   {OT1}{cmr} {m}{n}
\DeclareSymbolFont{letters}     {OML}{cmm} {m}{it}
\DeclareSymbolFont{symbols}     {OMS}{cmsy}{m}{n}
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}
\SetSymbolFont{letters}  {bold}{OML}{cmm} {b}{it}
\SetSymbolFont{symbols}  {bold}{OMS}{cmsy}{b}{n}
```

Below are the seven math alphabets which are defined by NFSS. Again they must be defined by this file. However, as before you can change the fonts used without losing portability, but you should be careful when changing the encoding since that may make documents come out wrong.

```
\DeclareSymbolFontAlphabet{\mathrm}    {operators}
\DeclareSymbolFontAlphabet{\mathnormal}{letters}
\DeclareSymbolFontAlphabet{\mathcal}   {symbols}
\DeclareMathAlphabet      {\mathbf}{OT1}{cmr}{bx}{n}
\DeclareMathAlphabet      {\mathsf}{OT1}{cmss}{m}{n}
\DeclareMathAlphabet      {\mathit}{OT1}{cmr}{m}{it}
\DeclareMathAlphabet      {\mathtt}{OT1}{cmtt}{m}{n}
```

Given the currently available fonts we cannot bold-en `\mathbf` and `\mathtt` but in principle one could use 'ultra bold' or something. The alphabets defined via `\DeclareSymbolFontAlphabet` will change automatically in a new math version if the corresponding symbol font changes.

```
\SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
\SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}
```

18.3 Math font sizes

For every *text* size NFSS maintains also three sizes that are used to typeset formulas.

The declarations below declare the text, script and scriptscript size to be used for each text font size.

All occurrences of sizes longer than a single character are replaced with the macro name that holds them, saving a number of tokens (but losing a bit of speed, so this may not stay this way).

```
 \DeclareMathSizes{5}{5}{5}{5}
 \DeclareMathSizes{6}{6}{5}{5}
 \DeclareMathSizes{7}{7}{5}{5}
 \DeclareMathSizes{8}{8}{6}{5}
 \DeclareMathSizes{9}{9}{6}{5}
 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
 \DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
 \DeclareMathSizes{\@xviipt}{\@xviipt}{\@xiipt}{\@xpt}
 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xviipt}
```

18.3.1 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.

```
\DeclareMathSymbol{a}{\mathalpha}{letters}{`a}
\DeclareMathSymbol{b}{\mathalpha}{letters}{`b}
\DeclareMathSymbol{c}{\mathalpha}{letters}{`c}
\DeclareMathSymbol{d}{\mathalpha}{letters}{`d}
\DeclareMathSymbol{e}{\mathalpha}{letters}{`e}
\DeclareMathSymbol{f}{\mathalpha}{letters}{`f}
\DeclareMathSymbol{g}{\mathalpha}{letters}{`g}
\DeclareMathSymbol{h}{\mathalpha}{letters}{`h}
\DeclareMathSymbol{i}{\mathalpha}{letters}{`i}
\DeclareMathSymbol{j}{\mathalpha}{letters}{`j}
\DeclareMathSymbol{k}{\mathalpha}{letters}{`k}
\DeclareMathSymbol{l}{\mathalpha}{letters}{`l}
\DeclareMathSymbol{m}{\mathalpha}{letters}{`m}
\DeclareMathSymbol{n}{\mathalpha}{letters}{`n}
\DeclareMathSymbol{o}{\mathalpha}{letters}{`o}
\DeclareMathSymbol{p}{\mathalpha}{letters}{`p}
\DeclareMathSymbol{q}{\mathalpha}{letters}{`q}
\DeclareMathSymbol{r}{\mathalpha}{letters}{`r}
\DeclareMathSymbol{s}{\mathalpha}{letters}{`s}
```

```
\DeclareMathSymbol{t}{\mathalpha}{letters}{`t}
\DeclareMathSymbol{u}{\mathalpha}{letters}{`u}
\DeclareMathSymbol{v}{\mathalpha}{letters}{`v}
\DeclareMathSymbol{w}{\mathalpha}{letters}{`w}
\DeclareMathSymbol{x}{\mathalpha}{letters}{`x}
\DeclareMathSymbol{y}{\mathalpha}{letters}{`y}
\DeclareMathSymbol{z}{\mathalpha}{letters}{`z}

\DeclareMathSymbol{A}{\mathalpha}{letters}{`A}
\DeclareMathSymbol{B}{\mathalpha}{letters}{`B}
\DeclareMathSymbol{C}{\mathalpha}{letters}{`C}
\DeclareMathSymbol{D}{\mathalpha}{letters}{`D}
\DeclareMathSymbol{E}{\mathalpha}{letters}{`E}
\DeclareMathSymbol{F}{\mathalpha}{letters}{`F}
\DeclareMathSymbol{G}{\mathalpha}{letters}{`G}
\DeclareMathSymbol{H}{\mathalpha}{letters}{`H}
\DeclareMathSymbol{I}{\mathalpha}{letters}{`I}
\DeclareMathSymbol{J}{\mathalpha}{letters}{`J}
\DeclareMathSymbol{K}{\mathalpha}{letters}{`K}
\DeclareMathSymbol{L}{\mathalpha}{letters}{`L}
\DeclareMathSymbol{M}{\mathalpha}{letters}{`M}
\DeclareMathSymbol{N}{\mathalpha}{letters}{`N}
\DeclareMathSymbol{O}{\mathalpha}{letters}{`O}
\DeclareMathSymbol{P}{\mathalpha}{letters}{`P}
\DeclareMathSymbol{Q}{\mathalpha}{letters}{`Q}
\DeclareMathSymbol{R}{\mathalpha}{letters}{`R}
\DeclareMathSymbol{S}{\mathalpha}{letters}{`S}
\DeclareMathSymbol{T}{\mathalpha}{letters}{`T}
\DeclareMathSymbol{U}{\mathalpha}{letters}{`U}
\DeclareMathSymbol{V}{\mathalpha}{letters}{`V}
\DeclareMathSymbol{W}{\mathalpha}{letters}{`W}
\DeclareMathSymbol{X}{\mathalpha}{letters}{`X}
\DeclareMathSymbol{Y}{\mathalpha}{letters}{`Y}
\DeclareMathSymbol{Z}{\mathalpha}{letters}{`Z}

\DeclareMathSymbol{0}{\mathalpha}{operators}{`0}
\DeclareMathSymbol{1}{\mathalpha}{operators}{`1}
\DeclareMathSymbol{2}{\mathalpha}{operators}{`2}
\DeclareMathSymbol{3}{\mathalpha}{operators}{`3}
\DeclareMathSymbol{4}{\mathalpha}{operators}{`4}
\DeclareMathSymbol{5}{\mathalpha}{operators}{`5}
\DeclareMathSymbol{6}{\mathalpha}{operators}{`6}
\DeclareMathSymbol{7}{\mathalpha}{operators}{`7}
\DeclareMathSymbol{8}{\mathalpha}{operators}{`8}
\DeclareMathSymbol{9}{\mathalpha}{operators}{`9}

\DeclareMathSymbol{!}{\mathclose}{operators}{"21}
\DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
\DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
\DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
\DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
\DeclareMathSymbol{.}{\mathord}{letters}{"3A}
\DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
\DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
\DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
\DeclareMathSymbol{?}{\mathclose}{operators}{"3F}
```

```
%\DeclareMathSymbol{(}{\mathopen}{operators}{"28}
%\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
%\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
%\DeclareMathSymbol{[}{\mathopen}{operators}{"5B}
%\DeclareMathSymbol{]}{\mathclose}{operators}{"5D}
%\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
%\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
%\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
```

```
%\DeclareMathSymbol{`\{}{\mathopen}{symbols}{"66}
%\DeclareMathSymbol{`\}}{\mathclose}{symbols}{"67}
%\DeclareMathSymbol{`\\}{\mathord}{symbols}{"6E} % \backslash
\mathcode`\ ="8000 % \space
\mathcode`\'="8000 % ^\prime
\mathcode`\_="8000 % \_
```

18.3.2 Delimitercodes for characters

```
\DeclareMathDelimiter{(}{\mathopen} ↩
      ↪{operators}{"28}{largesymbols}{"00}
\DeclareMathDelimiter{)}{\mathclose}{operators}{"29}{largesymbols}{"01}
\DeclareMathDelimiter{[}{\mathopen} ↩
      ↪{operators}{"5B}{largesymbols}{"02}
\DeclareMathDelimiter{]}{\mathclose}{operators}{"5D}{largesymbols}{"03}
```

The next two are considered to be relations when not used in the context of a delimiter! And worse, they do even represent different glyphs when being used as delimiter and not as delimiter. This is a user level syntax inherited from plain TEX. Therefore we explicitly redefine the math symbol definitions for these symbols afterwards.

```
\DeclareMathDelimiter{<}{\mathopen}{symbols}{"68}{largesymbols}{"0A}
\DeclareMathDelimiter{>}{\mathclose}{symbols}{"69}{largesymbols}{"0B}
\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
```

```
\DeclareMathDelimiter{/}{\mathord}{operators}{"2F}{largesymbols}{"0E}
\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
```

```
\DeclareMathDelimiter{|}{\mathord}{symbols}{"6A}{largesymbols}{"0C}
```

```
\expandafter\DeclareMathDelimiter\@backslashchar
                        {\mathord}{symbols}{"6E}{largesymbols}{"0F}
```

18.3.3 Symbols accessed via control sequences

18.3.4 Greek letters

```
\DeclareMathSymbol{\alpha}{\mathord}{letters}{"0B}
\DeclareMathSymbol{\beta}{\mathord}{letters}{"0C}
\DeclareMathSymbol{\gamma}{\mathord}{letters}{"0D}
\DeclareMathSymbol{\delta}{\mathord}{letters}{"0E}
\DeclareMathSymbol{\epsilon}{\mathord}{letters}{"0F}
\DeclareMathSymbol{\zeta}{\mathord}{letters}{"10}
```

```
\DeclareMathSymbol{\eta}{\mathord}{letters}{"11}
\DeclareMathSymbol{\theta}{\mathord}{letters}{"12}
\DeclareMathSymbol{\iota}{\mathord}{letters}{"13}
\DeclareMathSymbol{\kappa}{\mathord}{letters}{"14}
\DeclareMathSymbol{\lambda}{\mathord}{letters}{"15}
\DeclareMathSymbol{\mu}{\mathord}{letters}{"16}
\DeclareMathSymbol{\nu}{\mathord}{letters}{"17}
\DeclareMathSymbol{\xi}{\mathord}{letters}{"18}
\DeclareMathSymbol{\pi}{\mathord}{letters}{"19}
\DeclareMathSymbol{\rho}{\mathord}{letters}{"1A}
\DeclareMathSymbol{\sigma}{\mathord}{letters}{"1B}
\DeclareMathSymbol{\tau}{\mathord}{letters}{"1C}
\DeclareMathSymbol{\upsilon}{\mathord}{letters}{"1D}
\DeclareMathSymbol{\phi}{\mathord}{letters}{"1E}
\DeclareMathSymbol{\chi}{\mathord}{letters}{"1F}
\DeclareMathSymbol{\psi}{\mathord}{letters}{"20}
\DeclareMathSymbol{\omega}{\mathord}{letters}{"21}
\DeclareMathSymbol{\varepsilon}{\mathord}{letters}{"22}
\DeclareMathSymbol{\vartheta}{\mathord}{letters}{"23}
\DeclareMathSymbol{\varpi}{\mathord}{letters}{"24}
\DeclareMathSymbol{\varrho}{\mathord}{letters}{"25}
\DeclareMathSymbol{\varsigma}{\mathord}{letters}{"26}
\DeclareMathSymbol{\varphi}{\mathord}{letters}{"27}
\DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{"00}
\DeclareMathSymbol{\Delta}{\mathalpha}{operators}{"01}
\DeclareMathSymbol{\Theta}{\mathalpha}{operators}{"02}
\DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{"03}
\DeclareMathSymbol{\Xi}{\mathalpha}{operators}{"04}
\DeclareMathSymbol{\Pi}{\mathalpha}{operators}{"05}
\DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{"06}
\DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{"07}
\DeclareMathSymbol{\Phi}{\mathalpha}{operators}{"08}
\DeclareMathSymbol{\Psi}{\mathalpha}{operators}{"09}
\DeclareMathSymbol{\Omega}{\mathalpha}{operators}{"0A}
```

18.3.5 Ordinary symbols

```
\DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
\def\hbar{{\mathchar'26\mkern-9muh}}
\DeclareMathSymbol{\imath}{\mathord}{letters}{"7B}
\DeclareMathSymbol{\jmath}{\mathord}{letters}{"7C}
\DeclareMathSymbol{\ell}{\mathord}{letters}{"60}
\DeclareMathSymbol{\wp}{\mathord}{letters}{"7D}
\DeclareMathSymbol{\Re}{\mathord}{symbols}{"3C}
\DeclareMathSymbol{\Im}{\mathord}{symbols}{"3D}
\DeclareMathSymbol{\partial}{\mathord}{letters}{"40}
\DeclareMathSymbol{\infty}{\mathord}{symbols}{"31}
\DeclareMathSymbol{\prime}{\mathord}{symbols}{"30}
\DeclareMathSymbol{\emptyset}{\mathord}{symbols}{"3B}
\DeclareMathSymbol{\nabla}{\mathord}{symbols}{"72}
\def\surd{{\mathchar"1270}}
\DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
\DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
\def\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crcr
```

```
      \not\mathrel{\mkern14mu}\crcr
      \noalign{\nointerlineskip}
      \mkern2.5mu\leaders\hrule ↩
           ↪\@height.34pt\hfill\mkern2.5mu\crcr}}}}
\DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
\DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
\DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
\DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}
    \let\lnot=\neg
\DeclareMathSymbol{\flat}{\mathord}{letters}{"5B}
\DeclareMathSymbol{\natural}{\mathord}{letters}{"5C}
\DeclareMathSymbol{\sharp}{\mathord}{letters}{"5D}
\DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{"7C}
\DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{"7D}
\DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{"7E}
\DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{"7F}
```

18.3.6 Large Operators

```
\DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{"60}
\DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{"57}
\DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{"56}
\DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{"55}
\DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{"54}
\DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{"53}
\DeclareMathSymbol{\intop}{\mathop}{largesymbols}{"52}
    \def\int{\intop\nolimits}
\DeclareMathSymbol{\prod}{\mathop}{largesymbols}{"51}
\DeclareMathSymbol{\sum}{\mathop}{largesymbols}{"50}
\DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{"4E}
\DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{"4C}
\DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{"4A}
\DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{"48}
    \def\oint{\ointop\nolimits}
\DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{"46}
\DeclareMathSymbol{\smallint}{\mathop}{symbols}{"73}
```

18.3.60 Binary symbols

```
\DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{"2F}
\DeclareMathSymbol{\triangleright}{\mathbin}{letters}{"2E}
\DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{"34}
\DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{"35}
   \let \varbigtriangledown \bigtriangledown
   \let \varbigtriangleup \bigtriangleup
```

These last two synonyms are needed because the stamryrd package redefines them as Operators.

```
\DeclareMathSymbol{\wedge}{\mathbin}{symbols}{"5E}
   \let\land=\wedge
\DeclareMathSymbol{\vee}{\mathbin}{symbols}{"5F}
   \let\lor=\vee
\DeclareMathSymbol{\cap}{\mathbin}{symbols}{"5C}
\DeclareMathSymbol{\cup}{\mathbin}{symbols}{"5B}
```

```
\DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{"7A}
\DeclareMathSymbol{\dagger}{\mathbin}{symbols}{"79}
\DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{"75}
\DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{"74}
\DeclareMathSymbol{\uplus}{\mathbin}{symbols}{"5D}
\DeclareMathSymbol{\amalg}{\mathbin}{symbols}{"71}
\DeclareMathSymbol{\diamond}{\mathbin}{symbols}{"05}
\DeclareMathSymbol{\bullet}{\mathbin}{symbols}{"0F}
\DeclareMathSymbol{\wr}{\mathbin}{symbols}{"6F}
\DeclareMathSymbol{\div}{\mathbin}{symbols}{"04}
\DeclareMathSymbol{\odot}{\mathbin}{symbols}{"0C}
\DeclareMathSymbol{\oslash}{\mathbin}{symbols}{"0B}
\DeclareMathSymbol{\otimes}{\mathbin}{symbols}{"0A}
\DeclareMathSymbol{\ominus}{\mathbin}{symbols}{"09}
\DeclareMathSymbol{\oplus}{\mathbin}{symbols}{"08}
\DeclareMathSymbol{\mp}{\mathbin}{symbols}{"07}
\DeclareMathSymbol{\pm}{\mathbin}{symbols}{"06}
\DeclareMathSymbol{\circ}{\mathbin}{symbols}{"0E}
\DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{"0D}
\DeclareMathSymbol{\setminus}{\mathbin}{symbols}{"6E}
\DeclareMathSymbol{\cdot}{\mathbin}{symbols}{"01}
\DeclareMathSymbol{\ast}{\mathbin}{symbols}{"03}
\DeclareMathSymbol{\times}{\mathbin}{symbols}{"02}
\DeclareMathSymbol{\star}{\mathbin}{letters}{"3F}
\DeclareMathSymbol{\propto}{\mathrel}{symbols}{"2F}
\DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{"76}
\DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{"77}
\DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
\DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
\DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
\DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
\DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
\DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
\DeclareMathSymbol{\nwarrow}{\mathrel}{symbols}{"2D}
\DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
\DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
\DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
\DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
\def\neq{\not=} \let\ne=\neq
\DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
   \let\le=\leq
\DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}
   \let\ge=\geq
\DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
\DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
\DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
\DeclareMathSymbol{\succeq}{\mathrel}{symbols}{"17}
\DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
\DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
\DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
\DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
\DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
\DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
\DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}
```

```
    \let\owns=\ni
\DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
\DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
\DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
\DeclareMathSymbol{\leftrightarrow}{\mathrel}{symbols}{"24}
\DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
   \let\gets=\leftarrow
\DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}
   \let\to=\rightarrow
\DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
   \def\mapsto{\mapstochar\rightarrow}
\DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
\DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
\DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
\DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
\DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
\DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}
\DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
\DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
\DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
\DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
\DeclareMathSymbol{\rightharpoondown}{\mathrel}{letters}{"2B}
```

```
\DeclareRobustCommand
  \cong{\mathrel{\mathpalette\@vereq\sim}} % congruence sign
\def\@vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
    \ialign{$\m@th#1\hfil##\hfil$\crcr#2\crcr=\crcr}}}
\DeclareRobustCommand
  \notin{\mathrel{\m@th\mathpalette\c@ncel\in}}
\def\c@ncel#1#2{\m@th\ooalign{$\hfil#1\mkern1mu/\hfil$\crcr$#1#2$}}
\DeclareRobustCommand
  \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
\def\rlh@#1{\vcenter{\m@th\hbox{\ooalign{\raise2pt
          \hbox{$#1\rightharpoonup$}\crcr
        $#1\leftharpoondown$}}}}
\DeclareRobustCommand
  \doteq{\buildrel\textstyle.\over=}
```

18.3.60 Arrows

```
\DeclareRobustCommand
  \joinrel{\mathrel{\mkern-3mu}}
\DeclareRobustCommand
  \relbar{\mathrel{\smash-}} % \smash, because -
                               % has the same height as +
```

In contrast to `plain.tex` **`\Relbar`** got braces around the equal sign to guard against it being "math active" expanding to **`\futurelet...`**. This might be the case when packages are implementing shorthands for math, e.g. `=>` meaning **`\Rightarrow`** etc. It would actually be better not to use `=` in such definitions but instead define something like **`\mathequalsign`** and use this. However we can't do this now as it would break other math layouts where characters are in different places (since those wouldn't know about the need for a new command name).

```
\DeclareRobustCommand
  \Relbar{\mathrel{=}}
\DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
   \def\hookrightarrow{\lhook\joinrel\rightarrow}
\DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
   \def\hookleftarrow{\leftarrow\joinrel\rhook}
\DeclareRobustCommand
  \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
\DeclareRobustCommand
  \models{\mathrel{|}\joinrel\Relbar}
\DeclareRobustCommand
  \Longrightarrow{\Relbar\joinrel\Rightarrow}
\DeclareRobustCommand\longrightarrow
     {\relbar\joinrel\rightarrow}
\DeclareRobustCommand\longleftarrow
     {\leftarrow\joinrel\relbar}
\DeclareRobustCommand
  \Longleftarrow{\Leftarrow\joinrel\Relbar}
\DeclareRobustCommand
  \longmapsto{\mapstochar\longrightarrow}
\DeclareRobustCommand
  \longleftrightarrow{\leftarrow\joinrel\rightarrow}
\DeclareRobustCommand
  \Longleftrightarrow{\Leftarrow\joinrel\Rightarrow}
\DeclareRobustCommand
  \iff{\;\Longleftrightarrow\;}
```

18.3.60 Punctuation symbols

```
\DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
\DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
\DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}
%\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
%\DeclareRobustCommand\ldots
%          ↊
       ↋{\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots\,$}\fi}
\DeclareRobustCommand
  \cdots{\mathinner{\cdotp\cdotp\cdotp}}
\DeclareRobustCommand
  \vdots{\vbox{\baselineskip4\p@ \lineskiplimit\z@
    \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
\DeclareRobustCommand
  \ddots{\mathinner{\mkern1mu\raise7\p@
    \vbox{\kern7\p@\hbox{.}}\mkern2mu
    \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}
```

18.3.60 Math accents

```
\DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
\DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}
\DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
```

```
\DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
\DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
\DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
\DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
\DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
\DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
\DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
\DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
\DeclareMathAccent{\widehat}{\mathord}{largesymbols}{"62}
```

For some reason plain TEX never bothered to provide a ring accent in math (although it is available in the fonts), but since we got a request for it here we go:

```
\DeclareMathAccent{\mathring}{\mathalpha}{operators}{"17}
```

18.3.60 Radicals

```
\DeclareMathRadical{\sqrtsign}{symbols}{"70}{largesymbols}{"70}
```

18.3.60 Over and under something, etc

```
\def\overrightarrow#1{\vbox{\m@th\ialign{##\crcr
      \rightarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\def\overleftarrow#1{\vbox{\m@th\ialign{##\crcr
      \leftarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\def\overbrace#1{\mathop{\vbox{\m@th\ialign{##\crcr\noalign{\kern3\p@}%
      \downbracefill\crcr\noalign{\kern3\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}\limits}
\def\underbrace#1{\mathop{\vtop{\m@th\ialign{##\crcr
   $\hfil\displaystyle{#1}\hfil$\crcr
   \noalign{\kern3\p@\nointerlineskip}%
   \upbracefill\crcr\noalign{\kern3\p@}}}}\limits}
\def\skew#1#2#3{{\muskip\z@#1mu\divide\muskip\z@\tw@ ⤸
      ⤹\mkern\muskip\z@
    #2{\mkern-\muskip\z@{#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
\def\rightarrowfill{$\m@th\smash-\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
  \mkern-7mu\mathord\rightarrow$}
\def\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
  \mkern-7mu\smash-$}
\DeclareMathSymbol{\braceld}{\mathord}{largesymbols}{"7A}
\DeclareMathSymbol{\bracerd}{\mathord}{largesymbols}{"7B}
\DeclareMathSymbol{\bracelu}{\mathord}{largesymbols}{"7C}
\DeclareMathSymbol{\braceru}{\mathord}{largesymbols}{"7D}
\def\downbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
  \braceld\leaders\vrule \@height\ht\z@ ⤸
        ⤹\@depth\z@\hfill\braceru
  \bracelu\leaders\vrule \@height\ht\z@ ⤸
        ⤹\@depth\z@\hfill\bracerd$}
\def\upbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
```

```
  \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
  \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}
```

18.3.60 Delimiters

```
\DeclareMathDelimiter{\lmoustache}   % top from (, bottom from )
   {\mathopen}{largesymbols}{"7A}{largesymbols}{"40}
\DeclareMathDelimiter{\rmoustache}   % top from ), bottom from (
   {\mathclose}{largesymbols}{"7B}{largesymbols}{"41}
\DeclareMathDelimiter{\arrowvert}    % arrow without arrowheads
   {\mathord}{symbols}{"6A}{largesymbols}{"3C}
\DeclareMathDelimiter{\Arrowvert}    % double arrow without arrowheads
   {\mathord}{symbols}{"6B}{largesymbols}{"3D}
\DeclareMathDelimiter{\Vert}
   {\mathord}{symbols}{"6B}{largesymbols}{"0D}
\let\|=\Vert
\DeclareMathDelimiter{\vert}
   {\mathord}{symbols}{"6A}{largesymbols}{"0C}
\DeclareMathDelimiter{\uparrow}
   {\mathrel}{symbols}{"22}{largesymbols}{"78}
\DeclareMathDelimiter{\downarrow}
   {\mathrel}{symbols}{"23}{largesymbols}{"79}
\DeclareMathDelimiter{\updownarrow}
   {\mathrel}{symbols}{"6C}{largesymbols}{"3F}
\DeclareMathDelimiter{\Uparrow}
   {\mathrel}{symbols}{"2A}{largesymbols}{"7E}
\DeclareMathDelimiter{\Downarrow}
   {\mathrel}{symbols}{"2B}{largesymbols}{"7F}
\DeclareMathDelimiter{\Updownarrow}
   {\mathrel}{symbols}{"6D}{largesymbols}{"77}
\DeclareMathDelimiter{\backslash}    % for double coset G\backslash H
   {\mathord}{symbols}{"6E}{largesymbols}{"0F}
\DeclareMathDelimiter{\rangle}
   {\mathclose}{symbols}{"69}{largesymbols}{"0B}
\DeclareMathDelimiter{\langle}
   {\mathopen}{symbols}{"68}{largesymbols}{"0A}
\DeclareMathDelimiter{\rbrace}
   {\mathclose}{symbols}{"67}{largesymbols}{"09}
\DeclareMathDelimiter{\lbrace}
   {\mathopen}{symbols}{"66}{largesymbols}{"08}
\DeclareMathDelimiter{\rceil}
   {\mathclose}{symbols}{"65}{largesymbols}{"07}
\DeclareMathDelimiter{\lceil}
   {\mathopen}{symbols}{"64}{largesymbols}{"06}
\DeclareMathDelimiter{\rfloor}
   {\mathclose}{symbols}{"63}{largesymbols}{"05}
\DeclareMathDelimiter{\lfloor}
   {\mathopen}{symbols}{"62}{largesymbols}{"04}
```

```
\DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
     {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
\DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
     {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}
\DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
     {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}
```

18.3.7 Math versions of text commands

The **\mathunderscore** here is really a text definition, so it has been put back into **ltoutenc.dtx** (by Chris, 30/04/97) and should be removed from here.

These symbols are the math versions of text commands such as **\P**, **\$**, etc.

\mathparagraph **\mathsection** **\mathdollar** **\mathsterling** **\mathunderscore**

These math symbols are not in plain TEX.

\mathparagraph

```
\DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
```

\mathsection

The command produces § as compared to §.

```
\DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
```

```
\DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
\def\mathsterling{\mathit{\mathchar"7024}}
\def\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}
```

(*End definition for* `\mathparagraph` *and others.*)

\mathellipsis

This is plain TEX's **\ldots**.

```
\def\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%
```

(*End definition for* `\mathellipsis`.)

18.3.8 Other special functions and parameters

18.3.80 Bigggg

```
\def\big#1{{\hbox{$\left#1\vbox to8.5\p@{}\right.\n@space$}}}
\def\Big#1{{\hbox{$\left#1\vbox to11.5\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{}\right.\n@space$}}}
\def\n@space{\nulldelimiterspace\z@ \m@th}
```

18.3.9 The log-like functions

\operator@font

The **\operator@font** determines the symbol font used for log-like functions.

```
\def\operator@font{\mathgroup\symoperators}
```

18.3.10 Parameters

Lastly \thinmuskip, \medmuskip and \thickmuskip are set to initial values. It is not advisable to change these.

```
\thinmuskip=3mu
\medmuskip=4mu plus 2mu minus 4mu
\thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in fontmath.ltx.

```
%</math>
```

18.4 Default cfg files

We provide default cfg files here to ensure that on installations that search large file trees we do not pick up some strange customisation files from somewhere.

```
%<*cfgtext|cfgmath|cfgprel>
%%
%%
%%
%% Load the standard setup:
%%
%<+cfgtext>\input{fonttext.ltx}
%<+cfgmath>\input{fontmath.ltx}
%<+cfgprel>\input{preload.ltx}
%%
%% Small changes could go here; see documentation in cfgguide.tex for
%% allowed modifications.
%%
%% In particular it is not allowed to misuse this configuration file
%% to modify internal LaTeX commands!
%%
%% If you use this file as the basis for configuration please change
%% the \ProvidesFile lines to clearly identify your modification, e.g.,
%%
%<+cfgtext>%%  \ProvidesFile{fonttext.cfg}[2001/06/01
%<+cfgmath>%%  \ProvidesFile{fonttext.cfg}[2001/06/01
%<+cfgprel>%%   \ProvidesFile{preload.cfg}[2001/06/01
%%                                Customised local font setup]
%%
%%
%</cfgtext|cfgmath|cfgprel>
```

phd package and give a background as to how fonts are used in LaTeX.

19.1 Introduction

The preloading of fonts goes back to plain TEX and standardization.

> , it is desirable on many computer systems to preload the information for most of the fonts that people will actually be using, since this saves a lot of machine time [350].

19.2 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by LATEX 2_ε). It will be used to generate the following files:

preload.min	minimal subset of fonts necessary to run NFSS2
preload.ori	preload of CM fonts similar to the old `lfonts.tex`
preload.ltx	The standard selection of preloads
cmpreloa.xpt	preload of CM fonts for 10pt document size
cmpreloa.xip	preload of CM fonts for 11pt document size
cmpreloa.xii	preload of CM fonts for 12pt document size
dcpreloa.xpt	preload of DC fonts for 10pt size
dcpreloa.xip	preload of DC fonts for 11pt size
dcpreloa.xii	preload of DC fonts for 12pt size

These files are for installations that make use of Computer Modern fonts either old encoding (OT1) or Cork encoding (T1). The Computer Modern fonts with Cork encoding are known as DC-fonts.

Most important is `preload.ltx` which is used during format generation. You are *not* allowed to change this file.

19.3 Customization

You can customize the preloaded fonts in your LATEX 2_ε system by installing a file with the name `preload.cfg`. If this file exists it will be used in place of the system file `preload.ltx`. You can, for example, copy one of the files mentioned above (that can be generated from this source) to `preload.cfg`.

Or you can define completely other preloads. In that case start from `preload.min` since that contains the fonts that have to be preloaded by *all* LATEX 2_ε systems.

Avoid using `preload.ori`, it will load so many fonts that on most installations it is nearly impossible to load other font families afterwards. This file is only generated to show what fonts have been preloaded by LATEX 2.09.

If you normally use other fonts than Computer Modern `preload.min` might be best.

> **Warning:** If you preload fonts with encodings other than the normally supported encodings you have to declare that encoding in a

fontdef.cfg configuration file (see the documentation in the file fontdef.dtx). Adding an extra encoding to the format might produce non-portable documents, thus this should be avoided if possible.

19.4 Module switches for the DOCSTRIP program

The DOCSTRIP will generate the above file from this source using the following module directives:

driver	produce a documentation driver file
preload	produce a preload...file
cm	for OT1 encoded Computer Modern
dc	for T1 encoded Computer Modern
min	produce minimal subset
xpt	produce 10pt preloads
xipt	produce 11pt preloads
xiipt	produce 12pt preloads
ori	produce preloads similar to old lfonts.tex
tex	produce preload.ltx

A typical DOCSTRIP command file would then have entries like:

```
\generateFile{preload.min}{t}{\from{preload.dtx}{preload,min}}
```

for generating preload files.

19.5 A driver for this document

19.6 The code

We begin by loading the math extension font (cmex10) and the LaTeX line and circle fonts. It is necessary to do this explicitly since these are used by lplain.tex and latex.tex. Since the internal font name contains / characters and digits we construct the name via **\csname**. These are the only fonts (!) that must be loaded in this file.

All **\DeclarePreloadSizes** can be removed or others can be added, they only influence the processing speed.

```
\expandafter\font\csname OMX/cmex/m/n/10\endcsname=cmex10\relax
\font\tenln  =line10   \font\tenlnw  =linew10\relax
\font\tencirc=lcircle10 \font\tencircw=lcirclew10\relax
    \end{macrocode}
    The above fonts should not be touched but anything below this
    point here in the preload suggestions can be modified without any
    problems.
%    \begin{macrocode}
-tex%*****************************************
```

```
-tex% Start any modification below this point **
-tex%*****************************************
-tex
%%
%% Computer Modern Roman:
%%-----------------------
*ori
\begin{teX}
\DeclarePreloadSizes{OT1}{cmr}{m}{n}
        {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
\DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
\DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
\DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
\end{teX}
/ori
+xpt&cm \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
+xpt&dc \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
+xipt&cm \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
+xipt&dc \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
+xiipt&cm \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
+xiipt&dc \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
%%
%% Computer Modern Sans:
%%----------------------
+ori \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
%%
%% Computer Modern Typewriter:
%%----------------------------
+ori \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
%%
%% Computer Modern Math:
%%----------------------
*ori
\begin[teX}
\DeclarePreloadSizes{OML}{cmm}{m}{it}
         {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}
         {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
\end{teX}
/ori
```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```
%<*xpt>
\DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
%</xpt>
%<*xipt>
\DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
%</xipt>
%<*xiipt>
\DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
```

```
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
%</xiipt>
%%
%% LaTeX symbol fonts:
%%--------------------
%<*ori>
\DeclarePreloadSizes{U}{lasy}{m}{n}
         {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
%</ori>
%</preload>
```

phd package and give a background as to how fonts are used in LaTeX.

The commands defined in this file `ltfntcmd` *are part of the kernel code for LaTeX 2ε/NFSS2.*
It is also meant to serve as documentation for package writers since it demonstrates how to define high-level font changing commands using a small number of creator functions.

20.1 Introduction

Font changes such as **\bfseries**, **\sffamily**, etc. are declarations; this means that their scope is delimited by the grouping structure, either by the next **\end** of some environment or by explicitly using a group, e.g., writing something like **{\bfseries...}** in the source. If you make the mistake of writing **\bfseries{...}** (thinking of **\bfseries** as a command with one argument) then the result is rather striking.

Font declarations are an artifact of the TeX system and for several reasons it is better to avoid them on the user level whenever possible. In LaTeX3 they will probably all be replaced by environments and by font commands taking one argument.

This file defines a creator function for such declarative font switches. This function creates commands which can be used in both math and text.

This file also defines a number of high-level commands (all starting with **\text..**) that have one argument and typeset this argument in the requested way. Thus these commands are for typesetting short pieces of text in a specific family, series or shape. These are all produced as examples of the use of a creator function which is itself also defined in this file.

Table **??** 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**.[1] 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,
```

[1]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.

Table 20.1: Font-change commands with arguments

Command	*Corresponds to*	*Action*
`\textrm{..}`	`\rmfamily`	Typeset argument in roman family
`\textsf{..}`	`\sffamily`	Typeset argument in sans serif family
`\texttt{..}`	`\ttfamily`	Typeset argument in `typewriter` family
`\textmd{..}`	`\mdseries`	Typeset argument in medium series
`\textbf{..}`	`\bfseries`	Typeset argument in **bold** series
`\textup{..}`	`\upshape`	Typeset argument in normal shape
`\textit{..}`	`\itshape`	Typeset argument in *italic* shape
`\textsl{..}`	`\slshape`	Typeset argument in *slanted* shape
`\textsc{..}`	`\scshape`	Typeset argument in SMALL CAPS shape
`\emph{..}`	`\em`	Typeset argument *emphasized*

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.

```
the \emph{proper} use of italic corrections is
automatically taken care of}. Only
\emph{sometimes} one has to help \LaTeX{} by
adding a \verb=\nocorr= command.
```

which results in:

When using the NFSS high-level commands, the proper *use of italic corrections is automatically taken care of.* Only *sometimes* one has to help LaTeX by adding a `\nocorr` command.

In contrast, the use of the declaration forms is often more appropriate when you define your own commands or environments.

```
\newenvironment{bfitemize}{\begin{itemize}\normalfont\bfseries}
                          {\end{itemize}}
\begin{bfitemize}
\item This environment produces boldface items.
\item It is defined in terms of \LaTeX's
  \texttt{itemize} environment and NFSS
  declarations.
\end{bfitemize}
```

This gives:

- **This environment produces boldface items.**
- **It is defined in terms of LaTeX's `itemize` environment and NFSS declarations.**

In addition to global customization of when to insert the italic correction, it is of course sometimes necessary to explicitly insert one with `\/`.

It is also possible to suppress the italic correction in individual instances. For this, the command `\nocorr` is provided.

The `\nocorr` must appear as the first or last token inside the braces of the argument of the `\text...` commands, at that end of the text where you wish to suppress the italic correction.

It is worth pointing out here that inserting a `\/` in places where it can have no function (*i.e* anywhere except immediately after a slanted letter) is not an error—it will just be silently ignored. Unfortunately this is not true if the redefinition of `\/` in amstex.sty is used as this version can cause space to be removed immediately before the `\/`.

20.2 The implementation

`\DeclareTextFontCommand` {⟨*cmd*⟩}{⟨*code*⟩}

This is the creator function for `\text..` commands. It gives a warning if `\foo` or `\fragfoo` is already defined.

In math mode it simply puts the font declaration and text into a box (possibly an automagically sized one).

Otherwise it first scans the text to see where `\nocorr` occurs within it. This sets the `\check@ic` commands to do what is necessary concerning the italic correction at both ends.

The algorithm for deciding whether to put in an italic correction is not very subtle: one is added whenever the newly current font is not itself positively sloped, unless the next token is a character in the 'nocorr' list. At the end of the text this is done after closing the group so as to check the 'outer font'. Note that this will often result in adding an italic correction token after a character in an unsloped font; we believe (in early 2003) that this is perhaps inefficient but not dangerous.

It also now checks for empty contents of the text command and optimises this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other aftergroup tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```
\def \DeclareTextFontCommand #1#2{%
  \DeclareRobustCommand#1[1]{%
    \ifmmode
      \nfss@text{#2##1}%
```

```
    \else
      \hmode@bgroup
       \text@command{##1}%
       #2\check@icl ##1\check@icr
       \expandafter
      \egroup
    \fi
  }%
}
```

\textrm{⟨*text*⟩}

```
\DeclareTextFontCommand{\textrm}{\rmfamily}
\DeclareTextFontCommand{\textsf}{\sffamily}
\DeclareTextFontCommand{\texttt}{\ttfamily}
\DeclareTextFontCommand{\textnormal}{\normalfont}
\DeclareTextFontCommand{\textbf}{\bfseries}
\DeclareTextFontCommand{\textmd}{\mdseries}
\DeclareTextFontCommand{\textit}{\itshape}
\DeclareTextFontCommand{\textsl}{\slshape}
\DeclareTextFontCommand{\textsc}{\scshape}
\DeclareTextFontCommand{\textup}{\upshape}
\DeclareTextFontCommand{\emph}{\em}
\let \nocorr \relax
\let \check@icl \@empty
\let \check@icr \@empty
\def \text@command #1{%
  \def \reserved@a {#1}%
  \ifx \reserved@a \@empty
    \let \check@icl \@empty
    \let \check@icr \@empty
  \else
%    \def \reserved@b { }%
%    \ifx \reserved@a \reserved@b
    \ifx \reserved@a \space
      \let \check@icl \@empty
      \let \check@icr \@empty
    \else
      \check@nocorr@ #1\nocorr\@nil
    \fi
  \fi
}
\def \check@nocorr@ #1#2\nocorr#3\@nil {%
  \let \check@icl \maybe@ic
  \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi}%
  \def \reserved@a {\nocorr}%
  \def \reserved@b {#1}%
  \def \reserved@c {#3}%
  \ifx \reserved@a \reserved@b
    \ifx \reserved@c \@empty
```

```
      \let \check@icl \@empty
    \else
      \let \check@icl \@empty
      \let \check@icr \@empty
    \fi
  \else
    \ifx \reserved@c \@empty
    \else
      \let \check@icr \@empty
    \fi
  \fi
}
\newif\ifmaybe@ic
\def \maybe@ic {\futurelet\@let@token\maybe@ic@}
\def \maybe@ic@ {%
  \ifdim  \fontdimen\@ne\font>\z@
  \else
    \maybe@ictrue
    \expandafter\@tfor\expandafter\reserved@a\expandafter:\expandafter=%
        \nocorrlist
    \do \t@st@ic
    \ifmaybe@ic \sw@slant \fi
  \fi
}
\def \t@st@ic {%
  \expandafter\let\expandafter\reserved@b\expandafter=\reserved@a\relax
  \ifx\reserved@b\@let@token
    \maybe@icfalse
    \@break@tfor
  \fi
}
\def \sw@slant {%
  \ifdim \lastskip=\z@
    \fix@penalty
  \else
    \skip@ \lastskip
    \unskip
    \fix@penalty
    \hskip \skip@
  \fi
}
\def \fix@penalty {%
  \ifnum \lastpenalty=\z@
    \@@italiccorr
  \else
    \count@ \lastpenalty
    \unpenalty
```

```
    \@@italiccorr
    \penalty \count@
  \fi
}
\def \nocorrlist {,.}
\ifx \nfss@text\@undefined
  \def \nfss@text {\leavevmode\hbox}
\fi
\def \DeclareOldFontCommand #1#2#3{%
  \DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
}
\def \@fontswitch #1#2{%
  \ifmmode
     \let \math@bgroup \relax
     \def \math@egroup {\let \math@bgroup \@@math@bgroup
                        \let \math@egroup \@@math@egroup}%
     #2\relax
  \else
     #1%
  \fi
}
\let \@@math@bgroup \math@bgroup
\let \@@math@egroup \math@egroup
\@onlypreamble \DeclareTextFontCommand
\@onlypreamble \DeclareOldFontCommand
```

\normalsize

Initially this is defined to produce an error.

```
\def\normalsize{%
  \@latex@error {The font size command ↊
        ↋\protect\normalsize\space
              is not defined:\MessageBreak
              there is probably something wrong with
              the class file}\@eha
}
```

phd package and give a background as to how fonts are used in LaTeX.

21.1 Page Numbering

It is rather strange that this short piece of code is in its own dtx file. One would have expected it to perhaps be found in the file for counters. As expected 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`.

?? → P. **??** The user sets the pagenumber style with the `\pagenumbering`{⟨*foo*⟩} command, which sets the page counter to 1 and defines `\thepage` to be `\foo`. For example, `\pagenumbering{roman}` causes pages to be numbered i, ii, etc.

```
%<*2ekernel>
\message{page nos.,}
```

The `\cl@page` is set to empty. This is a list of counters to be reset, when the page is reset. If we had used the `\newcounter` control sequence from the Chapter on counters (see page **??**) this would have been created automatically.

```
\countdef\c@page=0 \c@page=1
\def\cl@page{}
```

`\pagenumbering`{⟨*name*⟩}

Sets the page numbering style. This will be called later in the output routine, when LaTeX is building up the page and getting it ready to be typeset.

```
\def\pagenumbering#1{%
  \global\c@page \@ne \gdef\thepage{\csname ⤸
       ⤹@#1\endcsname
   \c@page}}
```

```
%</2ekernel>
```

21.2 Modifications and additions by packages

The package **perpage** developed by David Kastrup adds the ability to reset counters per page and/or keep their occurences sorted in order of appearance on the page. It works by attaching itself to the code for **??** → P. **??** and will then modify the given counter according to information written to the `.aux` file, which means that multiple passes may be needed. Since it uses the internals of the echanism, the need for additional passes will get announced by LaTeX as "labels may have changed".

The package provides commands to get the absolute number of a counter (even if it is reset) by creating a command `c@abs`⟨*counter*⟩. For example if we call, `AddAbsoluteCounter{page}` it will create the counter `c@abspage`,

which can later be used as we require. This can be used for example to get the total pages of a document rather than the actual. The package **babel** might also interfere with counters.

phd package and give a background as to how fonts are used in LaTeX.

22.1 Cross Referencing

This section details the commands programmed in the LaTeX 2_ε kernel that deal with cross-referencing.

22.2 Author commands

`\label` The user writes `\label`{⟨*foo*⟩} to define a reference by a name. the name can be any character, where foo can be any string of characters not containing '`\`', '`{`' or '`}`'. cross-references:

The command will write to the `.aux` file something like this:

`\newlabel`{⟨*label*⟩}{⟨*{⟨currentlabel⟩}{⟨page⟩}*⟩}

where ⟨*label*⟩ is what is supplied by the user and currentlable is the current value of `\@currentlabel`, a macro that is usually updated whenever `refstepcounter` is called.

`\ref`

`\pageref`{⟨*name*⟩}

`\ref`{⟨*foo*⟩}: value of most recently incremented referencable counter. in the current environment. (Chapter, section, theorem and enumeration counters counters are referencable, footnote counters are not.)
`\pageref`{⟨*foo*⟩}: page number at which `\label{foo}` command appeared.
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. **??**

The author commands referenced in **??**

Most of this section is just a historical note, as newer documents inevitably use **hyperref**, which modifies the commands and the method of generation of references extensively. If you check the `aux` file you will probably find label commands as:

```
\newlabel{authorcommands}{{1}{5}{Cross Referencing}{section.2.1}{}}
```

22.2.1 Cross Referencing

```
982 \message{x-ref,}
```

This is implemented as follows. A referencable counter CNT is incremented by the command `\refstepcounter`{⟨*cnt*⟩} , which sets `\@currentlabel` == {⟨*CNT*⟩}{⟨*eval(*`\p@cnt\theCNT`*)*⟩}. The command `\label`{⟨*FOO*⟩} then writes the following on file `\@auxout` :

```
\newlabel{FOO}{{eval(\@currentlabel)}{eval(\thepage)}}}
```

```
\ref{⟨foo⟩} == begin
    if \r@foo undefined then
        @refundefined := G T
        ??
        Warning: 'reference foo on page ... undefined'
    else
        \@car \eval(\r@foo)\@nil
    end
end
\pageref{⟨foo⟩} = begin
    if \r@foo undefined then
        @refundefined := G T
        ??
        Warning: 'reference foo on page ... undefined'
    else
        \@cdr \eval(\r@foo)\@nil
    end
end
```

Algorithm 2: The referencing algorithm

`\@refundefined`

This does not save on name-space (since `\G@refundefinedfalse` was never needed) but it does make the implementation of such one-way switches more consistent. The extra macro to make the change is used since this change appears several times.

Note despite its name, `\G@refundefinedtrue` does *not* correspond to an `\if` command, and there is no matching ...`false`. It would be more natural to call the command `\G@refundefined` (as inspection of the change log will reveal) but unfortunately such a change would break any package that had defined a `\ref`-like command that mimicked the definition of `\ref`, calling `\G@refundefinedtrue`. Inspection of the TEX archives revealed several such packages, and so this command has been named ...`true` so that the definition of `\ref` need not be changed, and the packages will work without change..

```

 \newif\ifG@refundefined
 \def\G@refundefinedtrue{\global\let\ifG@refundefined\iftrue}
 \def\G@refundefinedfalse{\global\let\ifG@refundefined\iffalse}
\def\G@refundefinedtrue{%
  \gdef\@refundefined{%
    \@latex@warning@no@line{There were undefined references}}}
\let\@refundefined\relax
```

\ref

\@setref

made \ref and \pageref robust
Added setting of refundefined switch.

```
\def\@setref#1#2#3{%
  \ifx#1\relax
   \protect\G@refundefinedtrue
   \nfss@text{\reset@font\bfseries ??}%
   \@latex@warning{Reference `#3' on page \thepage \space
              undefined}%
  \else
   \expandafter#2#1\null
  \fi}

\def\ref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}}
\def\pageref#1{\expandafter\@setref\csname r@#1\endcsname
                                    \@secondoftwo{#1}}
```

\newlabel

Remove \@onlypreamble so still defined in new \enddocument This command will be written to the .aux file to pass label information from one run to another.

\@newl@bel

The internal form of \newlabel and \bibcite. Note that this macro does it's work inside a group. That way the local assignments it needs to do don't clutter the save stack. This prevents large documents with many labels to run out of save stack.

```
\def\@newl@bel#1#2#3{{%
  \@ifundefined{#1@#2}%
    \relax
    {\gdef \@multiplelabels {%
       \@latex@warning@no@line{There were multiply-defined labels}}%
     \@latex@warning@no@line{Label `#2' multiply defined}}%
  \global\@namedef{#1@#2}{#3}}}
\def\newlabel{\@newl@bel r}
\@onlypreamble\@newl@bel
```

\if@multiplelabels

\@multiplelabels

This is redefined to produce a warning if at least one label is defined more than once. It is executed by the \enddocument command.

```
\let \@multiplelabels \relax
```

`\label`

`\refstepcounter`

The commands `\label` and `\refstepcounter` have been changed to allow `\protect`'ed commands to work properly. For example,

```
\def\thechapter{\protect\foo{\arabic{chapter}.\roman{section}}}
```

will cause a `\label{bar}` command to define `\ref{bar}` to expand to something like `\foo{4.d}`.

```
\def\label#1{\@bsphack
  \protected@write\@auxout{}%
         {\string\newlabel{#1}{{\@currentlabel}{\thepage}}}%
  \@esphack}
```

```
\def\refstepcounter#1{\stepcounter{#1}%
    \protected@edef\@currentlabel
       {\csname p@#1\endcsname\csname the#1\endcsname}%
}
```

`\@currentlabel`

For `\label` commands that come before any environment

```
\def\@currentlabel{}
```

22.2.2 An extension of counter referencing

At the moment a reference to a counter `foo` will generate the equivalent of `\p@foo\thefoo` although not quite in this form. For some applications it would be nice of one could have `\thefoo` being an argument to `\p@foo` to be able to put material before and after the number generated by `\thefoo`. This can be easily achieved with a small change to one of the kernel commands as follows:

```
\def\refstepcounter#1{\stepcounter{#1}%
    \protected@edef\@currentlabel
       {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```

The trick is to ensure that `\csname the#1\endcsname` is turned into a single token before `\p@...` is expanded further. This way, if the `\p@...` command is a macro with one argument it will receive `\the...`. With the kernel code (i.e., without the `\expandafter`) it will instead pick up `\csname` which would be disastrous.

Using `\expandafter` instead of braces delimiting the argument is better because, assuming that the `\p@...` command is not defined as a macro with one argument, the braces will stay and prohibit kerning that might otherwise happen between the glyphs generated by `\the...` and surrounding glyphs.

We have refrained from making this change in the kernel code although for exisiting documents it would be 100% backward compatible. The reason being

that any class or package making use of this functionality would then horribly fail with older LaTeX installations.

Instead we suggest that people who are interested in using this functionality in a document class or package add the redefinition to the class file. To ensure that this redefinition is properly applied they might want to test for the original definition first, e.g.

```
\CheckCommand*\refstepcounter[1]{\stepcounter{#1}%
    \protected@edef\@currentlabel
       {\csname p@#1\endcsname\csname the#1\endcsname}%
}
\renewcommand*\refstepcounter[1]{\stepcounter{#1}%
    \protected@edef\@currentlabel
       {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```

phd package and give a background as to how fonts are used in LaTeX.

```
%<*2ekernel>
\message{environments,}
```

23.1 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.

`\@currenvir`⟨*environment name*⟩

The name of the current environment. Initialized to `document` to so that `\end{document}` works correctly.

```
\def\@currenvir{document}
```

```
\if@ignore
\@ignoretrue
\@ignorefalse
```

```
\def\@ignorefalse{\global\let\if@ignore\iffalse}
\def\@ignoretrue {\global\let\if@ignore\iftrue}
\@ignorefalse
```

(*End definition for* `\if@ignore`, `\@ignoretrue`, *and* `\@ignorefalse`.)

```
\ignorespacesafterend
```

(*End definition for* `\ignorespacesafterend`.)

```
\let\ignorespacesafterend\@ignoretrue
```

`\enddocument`

```
\def\enddocument{%
```

The `\end{document}` hook is executed first. If necessary it can contain a `\clearpage` to output dangling floats first. In this position it can also contain something like `\end{foo}` so that the whole document effectively starts and ends with some special environment. However, this must be used with care, eg if two applications would use this without knowledge of each other the order of the environments will be wrong after all. `\AtEndDocument` is redefined at this point so that and such commands that get into the hook do not chase their tail...

```
  \let\AtEndDocument\@firstofone
  \@enddocumenthook
  \@checkend{document}%
  \clearpage
  \begingroup
    \if@filesw
      \immediate\closeout\@mainaux
      \let\@setckpt\@gobbletwo
      \let\@newl@bel\@testdef
```

We use **\@@input** to load the `.aux` file, so that it doesn't show up in the list of files produced by **\listfiles**.

```
      \@tempswafalse
      \makeatletter \@@input\jobname.aux
    \fi
```

```
    \@dofilelist
```

First we check for font size substitution bigger than not a register.

```
    \ifdim \font@submax >\fontsubfuzz\relax
```

In case you wonder about the **\@gobbletwo** inside the message below, this is a horrible hack to remove the tokens **\on@line.** that are added by **\@font@warning** at the end.

```
      \@font@warning{Size substitutions with ↩
            ↪differences\MessageBreak
                  up to \font@submax\space have ↩
                        ↪occurred.\@gobbletwo}%
    \fi
```

The macro **\@defaultsubs** is initially **\relax** but gets redefined to produce a warning if there have been some default font substitutions.

```
    \@defaultsubs
```

The macro **\@refundefined** is initially **\relax** but gets redefined to produce a warning if there are undefined refs.

```
    \@refundefined
```

If a label is defined more than once, **\@tempswa** will always be true and thus produce a "Label(s) may ..." warning. But since a rerun will not solve that problem (unless one uses a package like `varioref` that generates labels on the fly), we suppress this message.

```
    \if@filesw
      \ifx \@multiplelabels \relax
        \if@tempswa
          \@latex@warning@no@line{Label(s) may have changed.
              Rerun to get cross-references right}%
        \fi
      \else
        \@multiplelabels
      \fi
    \fi
  \endgroup
  \deadcycles\z@\@@end}
```

\@testdef

```
\def\@testdef #1#2#3{%
  \def\reserved@a{#3}\expandafter \ifx \csname #1@#2\endcsname
 \reserved@a  \else \@tempswatrue \fi}
```

(*End definition for* `\@testdef`.)

\@writefile

```
\long\def\@writefile#1#2{%
  \@ifundefined{tf@#1}\relax
    {\@temptokena{#2}%
     \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
    }%
}
```

(*End definition for* `\@writefile`.)

\stop

```
\def\stop{\clearpage\deadcycles\z@\let\par\@@par\@@end}
```

(*End definition for* `\stop`.)

```
\everypar{\@nodocument} %% To get an error if text appears ↲
      ↳before the
\nullfont               %% \begin{document}
```

\begin

The **\begin** command takes one argument, checks if the argument is defined as a command and provides code to remember the line number. It also captures the environment name.

```
\def\begin#1{%
  \@ifundefined{#1}%
    {\def\reserved@a{\@latex@error{Environment #1 ↲
          ↳undefined}\@eha}}%
    {\def\reserved@a{\def\@currenvir{#1}%
     \edef\@currenvline{\on@line}%
     \csname #1\endcsname}}%
  \@ignorefalse
  \begingroup\@endpefalse\reserved@a}
```

Example 1.1 Environment examples

```
10 \newenvironment{mytest}{}{}
11 \begin{mytest}
12 \makeatletter
13 % Typesets mytest
14 Environment name: \@currenvir\\
15 % Typesets on input line 7
16 Environment line: \on@line
```

```

\ExplSyntaxOn
\cs_generate_variant:Nn \str_if_eq:nnTF {xnTF}
\str_if_eq:xnTF {\@currenvir}{mytest}{\PASS}{\FAIL}
\ExplSyntaxOff
\makeatother
\end{mytest}
```

Environment name: mytest
Environment line: on input line 7
PASS

`\end` The definition of `\end` is straight forward. It defines the end command for the environment such as `\endmyenvironment`. Checks that we ar ein the same environment, if not will issue @badend error.

```
\def\end#1{%
  \csname end#1\endcsname\@checkend{#1}%
  \expandafter\endgroup\if@endpe\@doendpe\fi
  \if@ignore\@ignorefalse\ignorespaces\fi}
```

(*End definition for* `\end`.)

`\@checkend`

```
\def\@checkend#1{\def\reserved@a{#1}\ifx
      \reserved@a\@currenvir \else\@badend{#1}\fi}
```

(*End 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. [1]

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

(*End definition for* `\@currenvline`.)

23.2 Center, Flushright, Flushleft

```
\message{center,}
```

`\@centercr`

```
\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
       \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
```

(*End definition for* `\@centercr`.)

[1] v1.0q 1994/05/24 Use `\@empty` as outer default

\@xcentercr

```
\def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
    [\@icentercr\ignorespaces}
```

(*End definition for* `\@xcentercr`.)

\@icentercr

```
\def\@icentercr[#1]{\vskip #1\ignorespaces}
```

(*End definition for* `\@icentercr`.)

center We use `\relax` to prevent `\item` scanning too far.

```
\def\center{\trivlist \centering\item\relax}
\def\endcenter{\endtrivlist}
```

\centering

```
\def\centering{%
  \let\\\@centercr
  \rightskip\@flushglue\leftskip\@flushglue
  \parindent\z@\parfillskip\z@skip}
```

(*End definition for* `\centering`.)

\@rightskip

```
\newskip\@rightskip \@rightskip \z@skip
```

(*End definition for* `\@rightskip`.)

```
\def\flushleft{\trivlist \raggedright\item\relax}
\def\endflushleft{\endtrivlist}
```

\raggedright

```
\def\raggedright{%
  \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
  \leftskip\z@skip
  \parindent\z@}
```

(*End definition for* `\raggedright`.)

```
\def\flushright{\trivlist \raggedleft\item\relax}
\def\endflushright{\endtrivlist}
```

\raggedleft

```
\def\raggedleft{%
  \let\\\@centercr
  \rightskip\z@skip\leftskip\@flushglue
  \parindent\z@\parfillskip\z@skip}
```

(*End definition for* `\raggedleft`.)

```
\message{verbatim,}
```

23.3 Verbatim

The verbatim environment uses the fixed-width **\ttfamily** font, turns blanks into spaces, starts a new line for each carriage return (or sequence of consecutive carriage returns), and interprets *every* character literally. I.e., all special characters **\, {, $**, etc. are **\catcode**'d to 'other'.

The command **\verb** produces in-line verbatim text, where the argument is delimited by any pair of characters. E.g., **\verb #...#** takes '**...**' as its argument, and sets it verbatim in **\ttfamily** font.

The *-variants of these commands are the same, except that spaces print as the TEXbook's space character instead of as blank spaces.

\@vobeyspaces

```
{\catcode`\ =\active%
\gdef\@vobeyspaces{\catcode`\ \active\let \@xobeysp}}
```

(*End definition for* `\@vobeyspaces`.)

\@xobeysp

(*End definition for* `\@xobeysp`.)

\@xverbatim
\@sxverbatim

```
\begingroup \catcode `|=0 \catcode `[= 1
\catcode`]=2 \catcode `\{=12 \catcode `\}=12
\catcode`\\=12 ↩
      ↳|gdef|@xverbatim#1\end{verbatim}[#1|end[verbatim]]
|gdef|@sxverbatim#1\end{verbatim*}[#1|end[verbatim*]]
|endgroup
```

(*End definition for* `\@xverbatim` *and* `\@sxverbatim`.)

\@verbatim

```
\def\@verbatim{\trivlist \item\relax
  \if@minipage\else\vskip\parskip\fi
  \leftskip\@totalleftmargin\rightskip\z@skip
  \parindent\z@\parfillskip\@flushglue\parskip\z@skip
```

Added **\@@par** to clear possible **\parshape** definition from a surrounding list (the verbatim guru says).

```
  \@@par
  \@tempswafalse
  \def\par{%
    \if@tempswa
```

A **\leavevmode** added: needed if, for example, a blank verbatim line is the first thing in a list item (wow!).

```
      \leavevmode \null \@@par\penalty\interlinepenalty
    \else
      \@tempswatrue
      \ifhmode\@@par\penalty\interlinepenalty\fi
    \fi}%
```

To allow customization we hide the font used in a separate macro. [2] [3] [4] [5]

```
  \let\do\@makeother \dospecials
  \obeylines \verbatim@font \@noligs
  \hyphenchar\font\m@ne
```

To avoid a breakpoint after the labels box, we remove the penalty put there by the list macros: another use of **\unpenalty**!

```
  \everypar \expandafter{\the\everypar \unpenalty}%
}
```

(*End definition for* `\@verbatim`.)

\verbatim

\endverbatim

```
\def\verbatim{\@verbatim \frenchspacing\@vobeyspaces ↱
      ↳\@xverbatim}
\def\endverbatim{\if@newlist \leavevmode\fi\endtrivlist}
```

\verbatim@font

```
\def\verbatim@font{\normalfont\ttfamily}
```

(*End definition for* `\verbatim@font`.)

```
\@namedef{verbatim*}{\@verbatim\@sxverbatim}
\expandafter\let\csname endverbatim*\endcsname =\endverbatim
```

\@makeother

```
\def\@makeother#1{\catcode`#112\relax}
```

(*End definition for* `\@makeother`.)

\verb@balance@group

```
\let\verb@balance@group\@empty
```

(*End definition for* `\verb@balance@group`.)

\verb@egroup

```
\def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
```

(*End definition for* `\verb@egroup`.)

\verb@eol@error

```
\begingroup
  \obeylines%
  \gdef\verb@eol@error{\obeylines%
    \def^^M{\verb@egroup\@latex@error{%
            \noexpand\verb ended by end of line}\@ehc}}%
\endgroup
```

[2] v0.9a,1993/11/21,use **\verbatim@font** instead of **\tt**.

[3] v0.9h,1993/12/13,Readded **\@noligs**.

[4] v1.1d,1996/06/03,Exchanged the following two code lines so that **\dospecials** cannot reset the category code of characters handled by **\@noligs**..

[5] v1.1h,2000/01/07,Disable hyphenation even if the font allows it..

(*End definition for* `\verb@eol@error`.)

`\verb`

```
\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
  \bgroup
    \verb@eol@error \let\do\@makeother \dospecials
    \verbatim@font\@noligs
    \@ifstar\@sverb\@verb}
```

(*End definition for* `\verb`.)

`\@sverb`

```
\def\@sverb#1{%
  \catcode`#1\active
  \lccode`\~`#1%
  \gdef\verb@balance@group{\verb@egroup
     \@latex@error{\noexpand\verb illegal in command ↩
           ↪argument}\@ehc}%
  \aftergroup\verb@balance@group
  \lowercase{\let~\verb@egroup}}%
```

(*End definition for* `\@sverb`.)

`\@verb`

```
\def\@verb{\@vobeyspaces \frenchspacing \@sverb}
```

(*End definition for* `\@verb`.)

`\verbatim@nolig@list`

```
\def\verbatim@nolig@list{\do\`\do\<\do\>\do\,\do\'\do\-}
```

(*End definition for* `\verbatim@nolig@list`.)

`\do@noligs`

```
\def\do@noligs#1{%
  \catcode`#1\active
  \begingroup
     \lccode`\~`#1\relax
     \lowercase{\endgroup\def~{\leavevmode\kern\z@\char`#1}}}
```

(*End definition for* `\do@noligs`.)

`\@noligs`

```
\def\@noligs{\let\do\do@noligs \verbatim@nolig@list}
```

(*End definition for* `\@noligs`.)

phd package and give a background as to how fonts are used in LaTeX.

This module is in **kernel-ltmath.dtx**. The file provides a number of math environments, and for the rest is mostly a copy of plain.

24.1 Math commands based on plain TEX

24.1.1 The log-like functions

Note that `\operator@font` is defined in the **fontdef.dtx** file (see chapter **??**).

The standard operators:

```
\def\log{\mathop{\operator@font log}\nolimits}
\def\lg{\mathop{\operator@font lg}\nolimits}
\def\ln{\mathop{\operator@font ln}\nolimits}
\def\lim{\mathop{\operator@font lim}}
\def\limsup{\mathop{\operator@font lim\,sup}}
\def\liminf{\mathop{\operator@font lim\,inf}}
\def\sin{\mathop{\operator@font sin}\nolimits}
\def\arcsin{\mathop{\operator@font arcsin}\nolimits}
\def\sinh{\mathop{\operator@font sinh}\nolimits}
\def\cos{\mathop{\operator@font cos}\nolimits}
\def\arccos{\mathop{\operator@font arccos}\nolimits}
\def\cosh{\mathop{\operator@font cosh}\nolimits}
\def\tan{\mathop{\operator@font tan}\nolimits}
\def\arctan{\mathop{\operator@font arctan}\nolimits}
\def\tanh{\mathop{\operator@font tanh}\nolimits}
\def\cot{\mathop{\operator@font cot}\nolimits}
\def\coth{\mathop{\operator@font coth}\nolimits}
\def\sec{\mathop{\operator@font sec}\nolimits}
\def\csc{\mathop{\operator@font csc}\nolimits}
\def\max{\mathop{\operator@font max}}
\def\min{\mathop{\operator@font min}}
\def\sup{\mathop{\operator@font sup}}
\def\inf{\mathop{\operator@font inf}}
\def\arg{\mathop{\operator@font arg}\nolimits}
\def\ker{\mathop{\operator@font ker}\nolimits}
\def\dim{\mathop{\operator@font dim}\nolimits}
\def\hom{\mathop{\operator@font hom}\nolimits}
\def\det{\mathop{\operator@font det}}
\def\exp{\mathop{\operator@font exp}\nolimits}
\def\Pr{\mathop{\operator@font Pr}}
\def\gcd{\mathop{\operator@font gcd}}
\def\deg{\mathop{\operator@font deg}\nolimits}
```

`\bmod`

Some of the operators are hand crafted, for modular exponentiation `\bmod` is defined as. Note that `\nonscript` suppresses the following space in the script styles.

$$c = (b \cdot c) \bmod m$$

```
\def\bmod{%
  \nonscript\mskip-\medmuskip\mkern5mu%
  \mathbin{\operator@font mod}\penalty900\mkern5mu%
  \nonscript\mskip-\medmuskip}
```

\pmod

```
\def\pmod#1{%
  \allowbreak\mkern18mu({\operator@font mod}\,\,#1)}
```

24.1.2 Delimiters Bigggg

Note that **\big** and similar commands are defined by the **fontmath.dtx** module.

\big

Variants on **\big** and friends for use with delimiters:

```
\def\bigl{\mathopen\big}
\def\bigm{\mathrel\big}
\def\bigr{\mathclose\big}
\def\Bigl{\mathopen\Big}
\def\Bigm{\mathrel\Big}
\def\Bigr{\mathclose\Big}
\def\biggl{\mathopen\bigg}
\def\biggm{\mathrel\bigg}
\def\biggr{\mathclose\bigg}
\def\Biggl{\mathopen\Bigg}
\def\Biggm{\mathrel\Bigg}
\def\Biggr{\mathclose\Bigg}
```

24.1.20 The UNSORTED Rest

The other math commands are lifted from plain TEX.

\jot

```
\newdimen\jot
\jot=3pt
```

\interdisplaylinepenalty

```
\newcount\interdisplaylinepenalty
\interdisplaylinepenalty=100
```

\choose

```
\def\choose{\atopwithdelims()}
```

\brack

```
\def\brack{\atopwithdelims[]}
```

\brace

```
\def\brace{\atopwithdelims\{\}}
```

\mathpalette

```
\def\mathpalette#1#2{%
  \mathchoice
    {#1\displaystyle{#2}}%
    {#1\textstyle{#2}}%
    {#1\scriptstyle{#2}}%
    {#1\scriptscriptstyle{#2}}}
```

\root

\rootbox

\r@@t

```
\newbox\rootbox
```

```
\def\root#1\of{%
  \setbox\rootbox\hbox{$\m@th\scriptscriptstyle{#1}$}%
  \mathpalette\r@@t}
```

```
\def\r@@t#1#2{%
  \setbox\z@\hbox{$\m@th#1\sqrtsign{#2}$}%
  \dimen@\ht\z@ \advance\dimen@-\dp\z@
  \mkern5mu\raise.6\dimen@\copy\rootbox
  \mkern-10mu\box\z@}
```

\phantom

\hphantom

\vphantom

```
\newif\ifv@
\newif\ifh@
```

```
\def\vphantom{\v@true\h@false\ph@nt}
```

```
\def\hphantom{\v@false\h@true\ph@nt}
```

```
\def\phantom{\v@true\h@true\ph@nt}
```

```
\def\ph@nt{%
  \ifmmode
    \expandafter\mathpalette\expandafter\mathph@nt
  \else
    \expandafter\makeph@nt
  \fi}
```

```
\def\makeph@nt#1{%
  \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finph@nt}
\def\mathph@nt#1#2{%
  \setbox\z@\hbox{$\m@th#1{#2}$}\finph@nt}
\def\finph@nt{%
  \setbox\tw@\null
  \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
  \ifh@ \wd\tw@\wd\z@\fi \box\tw@}
```

\mathstrut

```
\def\mathstrut{\vphantom(}
```

\smash

```
\def\smash{%
  \relax % \relax, in case this comes first in \halign
  \ifmmode
    \expandafter\mathpalette\expandafter\mathsm@sh
  \else
    \expandafter\makesm@sh
  \fi}
\def\makesm@sh#1{%
  \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
\def\mathsm@sh#1#2{%
  \setbox\z@\hbox{$\m@th#1{#2}$}\finsm@sh}
\def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}
```

\buildrel

```
\def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}
```

\cases

```
\def\cases#1{\left\{\,\vcenter{\normalbaselines\m@th
    \ialign{$##\hfil$&\quad{##}\hfil\crcr#1\crcr}}\right.}
```

$$\left\{\begin{array}{ll} x & \text{if } x0, \\ -x & \text{if } x \leq 0. \end{array}\right.$$

\matrix

```
\def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
    \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crcr
      \mathstrut\crcr\noalign{\kern-\baselineskip}
      #1\crcr\mathstrut\crcr\noalign{\kern-\baselineskip}}}\,}
```

\pmatrix

```
\def\pmatrix#1{\left(\matrix{#1}\right)}
```

$$M = \begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

\bordermatrix

The bordermatrix

```
\def\bordermatrix#1{\begingroup \m@th
  \@tempdima 8.75\p@
  \setbox\z@\vbox{%
    \def\cr{\crcr\noalign{\kern2\p@\global\let\cr\endline}}%
    \ialign{$##$\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
      &&\quad\hfil$##$\hfil\crcr
      \omit\strut\hfil\crcr\noalign{\kern-\baselineskip}%
      #1\crcr\omit\strut\cr}}%
  \setbox\tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
  \setbox\tw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
  \setbox\tw@\hbox{$\kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
    \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
    \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,\right)$}%
  \null\;\vbox{\kern\ht\@ne\box\tw@}\endgroup}
```

\openup

```
\def\openup{\afterassignment\@penup\dimen@}
```

```
\def\@penup{\advance\lineskip\dimen@
  \advance\baselineskip\dimen@
  \advance\lineskiplimit\dimen@}
```

\displaylines

```
\newif\ifdt@p
```

```
\def\displ@y{\global\dt@ptrue\openup\jot\m@th
  \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
      \vskip-\lineskiplimit \vskip\normallineskiplimit \fi
      \else \penalty\interdisplaylinepenalty \fi}}}
```

```
\def\@lign{\tabskip\z@skip\everycr{}} % restore inside \displ@y
```

```
\def\displaylines#1{\displ@y \tabskip\z@skip
  \halign{\hb@xt@\displaywidth{$\@lign\hfil\displaystyle##\hfil$}\crcr
    #1\crcr}}
```

\sp

\sb

```
\let\sp=^
\let\sb=_
```

\>

\;

\!

```
%\def\,{\mskip\thinmuskip}      % already defined in ltspace
\def\>{\mskip\medmuskip}
\def\;{\mskip\thickmuskip}
\def\!{\mskip-\thinmuskip}
```

*

```
\def\*{\discretionary{\thinspace\the\textfont2\char2}{}{}}
```

\:

```
\let\:=\>
```

\active@math@prime

```
\def\active@math@prime{^\bgroup\prim@s}
```

\prime@s

```
{\catcode`\'=\active \global\let'\active@math@prime}
```

```
\def\prim@s{%
  \prime\futurelet\@let@token\pr@m@s}
```

```
\def\pr@m@s{%
  \ifx'\@let@token
    \expandafter\pr@@@s
  \else
    \ifx^\@let@token
      \expandafter\expandafter\expandafter\pr@@@t
    \else
      \egroup
    \fi
  \fi}
```

```
\def\pr@@@s#1{\prim@s}
```

```
\def\pr@@@t#1#2{#2\egroup}
```

```
{\catcode`\_=\active \gdef_{\_}} % _ in math is
                                 % either subscript or \_
```

\(

\)

Produces `$...$` with checks that `\(` isn't used in math mode, and that `\)` is only used in math mode begun with `\(`.

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\(}{Make \( robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\({%
  \relax\ifmmode\@badmath\else$\fi}%
\DeclareRobustCommand\){%
  \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\(}{Make \( robust}%
%<latexrelease>\def\({%
%<latexrelease>  \relax\ifmmode\@badmath\else$\fi}%
%<latexrelease>\def\){%
%<latexrelease>  \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

```
\[

    \]
```

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}{\[}{Make \[ robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\[{%
   \relax\ifmmode
      \@badmath
   \else
      \ifvmode
         \nointerlineskip
         \makebox[.6\linewidth]{}%
      \fi
      $$%%$$ BRACE MATCH HACK
   \fi
}%
\DeclareRobustCommand\]{%
   \relax\ifmmode
      \ifinner
         \@badmath
      \else
         $$%%$$ BRACE MATCH HACK
      \fi
   \else
      \@badmath
   \fi
   \ignorespaces
}%
```

```
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}{\[}{Make \[ robust}%
%<latexrelease>\def\[{%
%<latexrelease>   \relax\ifmmode
%<latexrelease>      \@badmath
%<latexrelease>   \else
%<latexrelease>      \ifvmode
%<latexrelease>         \nointerlineskip
%<latexrelease>         \makebox[.6\linewidth]{}%
%<latexrelease>      \fi
%<latexrelease>      $$%%$$ BRACE MATCH HACK
%<latexrelease>   \fi
%<latexrelease>}%
%<latexrelease>\def\]{%
%<latexrelease>   \relax\ifmmode
%<latexrelease>      \ifinner
%<latexrelease>         \@badmath
%<latexrelease>      \else
%<latexrelease>         $$%%$$ BRACE MATCH HACK
%<latexrelease>      \fi
%<latexrelease>   \else
%<latexrelease>      \@badmath
%<latexrelease>   \fi
%<latexrelease>   \ignorespaces
%<latexrelease>}%
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

math

displaymath Disguises for `\(...\)` and `\[...\]`.

```
\let\math=\(
\let\endmath=\)
\def\displaymath{\[}
\def\enddisplaymath{\]\@ignoretrue}
```

equation The equation environment.

\c@equation

Numbered equations, using the counter `c@equation`. *Note*: The document style must define `\theequation` etc., and do the appropriate `\@addtoreset`. It should also redefine `\@eqnnum` if another format for the equation number is desired other than the standard (...), or to move the equation numbers to the flushleft. (See comment on the `\def` of `\@eqnnum`.)

```
\@definecounter{equation}
\def\equation{$$\refstepcounter{equation}}
\def\endequation{\eqno \hbox{\@eqnnum}$$\@ignoretrue}
```

\@eqnnum

```
\def\@eqnnum{{\normalfont \normalcolor (\theequation)}}
```

\stackrel

```
\def\stackrel#1#2{\mathrel{\mathop{#2}\limits^{#1}}}
```

\frac

```
\def\frac#1#2{{\begingroup#1\endgroup\over#2}}
```

\sqrt [⟨*nth root*⟩]

The **sqrt** macro extends the TEX macros to take an optional argument for the root. It uses the macros **\sqrtsign** and the contstruction **\root**⟨*value*⟩**\of**, through the auxiliary function **??**[→ P. ??]. All the commands are robust in LATEX 2015. The **\sqrtsign** is defined in the **mathdef.dtx** file

\@sqrt

Add an optional argument to plain's **\sqrt** to give the *n*th root of an expression $\sqrt[n]{e}$.

```
\DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}
\def\@sqrt[#1]{\root #1\of}
```

\@eqcnt

\@eqpen

\if@eqnsw

\@eqnsel

```
\newcount\@eqcnt
\newcount\@eqpen
\newif\if@eqnsw\@eqnswtrue
\newskip\@centering
\@centering = 0pt plus 1000pt
\def\eqnarray{%
   \stepcounter{equation}%
   \def\@currentlabel{\p@equation\theequation}%
   \global\@eqnswtrue
   \m@th
   \global\@eqcnt\z@
   \tabskip\@centering
   \let\\\@eqncr
   $$\everycr{}\halign to\displaywidth\bgroup
       \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsel
      &\global\@eqcnt\@ne\hskip \tw@\arraycolsep ⤹
           ⤷\hfil${##}$\hfil
      &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
         $\displaystyle{##}$\hfil\tabskip\@centering
      &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
         \tabskip\z@skip
      \cr
}
```

```
\def\endeqnarray{%
      \@@eqncr
      \egroup
      \global\advance\c@equation\m@ne
   $$\@ignoretrue
}
\let\@eqnsel=\relax
```

\nonumber

```
\def\nonumber{\global\@eqnswfalse}
```

\@eqncr

\@xeqncr

\@yeqncr

```
\def\@eqncr{%
   {\ifnum0=`}\fi
   \@ifstar{%
      \global\@eqpen\@M\@yeqncr
   }{%
      \global\@eqpen\interdisplaylinepenalty \@yeqncr
   }%
}
\def\@yeqncr{\@testopt\@xeqncr\z@skip}
\def\@xeqncr[#1]{%
   \ifnum0=`{\fi}%
   \@@eqncr
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
}

\def\@@eqncr{\let\reserved@a\relax
    \ifcase\@eqcnt \def\reserved@a{& & &}\or \def\reserved@a{& &}%
     \or \def\reserved@a{&}\else
       \let\reserved@a\@empty
       \@latex@error{Too many columns in eqnarray environment}\@ehc\fi
     \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
     \global\@eqnswtrue\global\@eqcnt\z@\cr}
```

\@seqncr

```
\let\@seqncr=\@eqncr
\@namedef{eqnarray*}{\def\@eqncr{\nonumber\@seqncr}\eqnarray}
\@namedef{endeqnarray*}{\nonumber\endeqnarray}
```

\lefteqn

```
\def\lefteqn#1{\rlap{$\displaystyle #1$}}
```

\ensuremath

```
\DeclareRobustCommand{\ensuremath}{%
  \ifmmode
    \expandafter\@firstofone
  \else
    \expandafter\@ensuredmath
  \fi}
```

\@ensuredmath

```
\long\def\@ensuredmath#1{$\relax#1$}
```

```
%</2ekernel>
```

24.1.20 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.

```
%<*leqno>
\renewcommand\@eqnnum{\hb@xt@.01\p@{}%
                      \rlap{\normalfont\normalcolor
                        \hskip -\displaywidth(\theequation)}}
%</leqno>
```

24.1.20 Flush left equations

To get the displayed math environments to print the contents flush left (with an indentation) we have to redefine all of LaTeX 2_ε's displayed math environments.

\mathindent{⟨*length*⟩}

The amount of indentation of the equations is stored in a register. Currently in this document it is

```
%<*fleqn>
\newdimen\mathindent
```

The setting of **\mathindent** has to be deferred until the class file has been processed, because **\leftmargini** is still opt wide at the moment `fleqn.clo` is read in.

```
\AtEndOfClass{\mathindent\leftmargini}
```

\[

The \[is defined here to begin display math. Note that in the later releases it has been made robust.

```
\IncludeInRelease{2015/01/01}{\[}{Make \[ robust}%
\DeclareRobustCommand\[{\relax
  \ifmmode\@badmath
  \else
  \begin{trivlist}%
     \@beginparpenalty\predisplaypenalty
     \@endparpenalty\postdisplaypenalty
     \item[]\leavevmode
            \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
            \hskip\mathindent\bgroup
  \fi}
\EndIncludeInRelease
```

Restore in earlier releases as:

```
\IncludeInRelease{0000/00/00}{\[}{Make \[ robust}%
\renewcommand\[{\relax
                \ifmmode\@badmath
                \else
                  \begin{trivlist}%
                    \@beginparpenalty\predisplaypenalty
                    \@endparpenalty\postdisplaypenalty
                    \item[]\leavevmode
                    \hb@xt@\linewidth\bgroup ↩
                         ↪$\m@th\displaystyle %$
                      \hskip\mathindent\bgroup
                \fi}
\EndIncludeInRelease
```

Example 1.1 Check if is robust

```
\meaning\[
```

\long macro:->\begin {linenomath}\LN@displaymath

\]

The end display math is defined in a similar fashion to the begin math command.

```
\IncludeInRelease{2015/01/01}{\]}{Make \] robust}%
\DeclareRobustCommand\]{\relax
                \ifmmode
                      \egroup $\hfil% $
                    \egroup
                  \end{trivlist}%
                \else \@badmath
                \fi}
\EndIncludeInRelease
```

```
\IncludeInRelease{0000/00/00}{\]}{Make \] robust}%
\renewcommand\]{\relax
                \ifmmode
                      \egroup $\hfil% $
                    \egroup
                  \end{trivlist}%
                \else \@badmath
                \fi}
\EndIncludeInRelease
```

\begin{equation}
⟨*environment content*⟩
\end{equation}

The equation environment

```
\renewenvironment{equation}%
    {\@beginparpenalty\predisplaypenalty
     \@endparpenalty\postdisplaypenalty
     \refstepcounter{equation}%
     \trivlist \item[]\leavevmode
       \hb@xt@\linewidth\bgroup $\m@th% $
         \displaystyle
         \hskip\mathindent}%
        {$\hfil % $
         \displaywidth\linewidth\hbox{\@eqnum}%
       \egroup
     \endtrivlist}
```

\begin{eqnarray}
⟨*environment content*⟩
\end{eqnarray}

```
\renewenvironment{eqnarray}{%
  \stepcounter{equation}%
  \def\@currentlabel{\p@equation\theequation}%
  \global\@eqnswtrue\m@th
  \global\@eqcnt\z@
  \tabskip\mathindent
  \let\\=\@eqncr
  \setlength\abovedisplayskip{\topsep}%
  \ifvmode
    \addtolength\abovedisplayskip{\partopsep}%
  \fi
```

When the documentclass uses a non-zero `\parskip` setting the `\topsep` might have a negative value to compensate for that. Therefore we add `\parskip` to `\abovedisplayskip`.

```
    \addtolength\abovedisplayskip{\parskip}%
    \setlength\belowdisplayskip{\abovedisplayskip}%
    \setlength\belowdisplayshortskip{\abovedisplayskip}%
    \setlength\abovedisplayshortskip{\abovedisplayskip}%
```

```
    $$\everycr{}\halign to\linewidth% $$
    \bgroup
      \hskip\@centering
      $\displaystyle\tabskip\z@skip{##}$\@eqnsel&%
      \global\@eqcnt\@ne \hskip \tw@\arraycolsep ⤸
            ⤹\hfil${##}$\hfil&%
      \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
        $\displaystyle{##}$\hfil \tabskip\@centering&%
      \global\@eqcnt\thr@@
        \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}%
      {\@@eqncr
    \egroup
    \global\advance\c@equation\m@ne$$% $$
    \@ignoretrue
    }
%</fleqn>
```

phd package and give a background as to how fonts are used in LaTeX.

25.1 List, and related environments

This section of the kernel is interesting in order to understand the underlying general design decisions of the LaTeX Team.

The kernel source can be considered as a framework, upon which classes build. All LaTeX lists are aware of their environment and can be included in other lists. You can also even if you want include sections within enumerated or itemized lists and they will not break. You can include boxes or minipages.

This section of the kernel creates generic factory commands for the creation of list environments. These list environments `\env:enumerate` `\env:itemize`re defined in the kernel, whereas others are normally defined in the class files (see **??**, **??** and **??**).

To understand how lists are created we need to understand that they are shaped paragraphs. Lamport's decision to shape them with `\parshape`, has produced a flexible system, used later on by standard classes to create numerous environments. The complexity of the code is necessary to make sure that at every item the `\parshape` is used and also in any following paragraphs. Remember that `\parshape` will only act on one paragraph. As Knuth said the LaTeX Team did some nifty tricks with `\parshape`.

A sideline is that the code can be used to add vertical and bottom space to content as well as adjust the left and right margins of text blocks. In the LaTeX classes they are used to define the environments for quotations, quotes **??**, verse **??** and verbatim environments.

25.2 Generic Command

The `\list` is the generic command that is used to create other lists. Think of it as the `\newcommand` for new list environments. The command takes two arguments.

`\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`[item label] 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. This enables the user to type `item` or `item[some 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 **??**$^{\rightarrow \text{P.}\,??}$ to have the default definition:

`\makelabel`{⟨*arg*⟩} == BEGIN `\hfil` arg END

which, for a label of width less than **??**$^{\rightarrow \text{P.}\,??}$, puts the label flushright, **??**$^{\rightarrow \text{P.}\,??}$ to the left of the item's text. However, `\makelabel` can be `\let` to another command by the **??**$^{\rightarrow \text{P.}\,??}$'s ⟨*commands*⟩ argument.

A **??**$^{\rightarrow \text{P.}??}$\{⟨*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 **??**$^{\rightarrow \text{P.}??}$ command.

Blank lines are ignored every other reasonable place–i.e.:

- Between the `\begin{list}` and the first `\item`,
- Between the `\item` and the text of that item.
- Between the end of the last item and the `\end{list}`.

For an environment like `quotation`, in which items are not labeled, the entire environment is a single item. It is defined by letting `\quotation` == `\list{}{...}\item\relax`. (Note the `\relax`, there in case the first character in the environment is a '['.) The spacing parameters provide a great deal of flexability in designing the format, including the ability to let the indentation of the first paragraph be different from that of the subsequent ones.

The trivlist environment is equivalent to a list environment whose second argument sets the following parameter values:

??$^{\rightarrow \text{P.}??}$ = 0: causes no indentation of left margin

??$^{\rightarrow \text{P.}??}$ **= 0:** see below for precise effect this has.

??$^{\rightarrow \text{P.}??}$ **= 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.

Example 2.1 understanding how lists are being build

```

\makeatletter
\def\mybullet{$\bullet$}
\leftskip0pt \rightskip0pt

\newlength\lmargin
\newlength\rmargin
\newlength\lnwidth
\setlength\lnwidth{\linewidth}
```

```latex

\setlength\lmargin{30pt}
\setlength\rmargin{30pt}
\setlength\lnwidth{\textwidth}
\advance\lnwidth-2\rmargin\relax %

\hskip-\lmargin\hbox to \lmargin{\hfill\mybullet}%
\parshape 1 \dimexpr\lmargin\relax \lnwidth\relax
\lorem

% Second list level
\addtolength\lmargin{30pt}\relax
\hskip-\lmargin\hbox to \lmargin{\hfill\mybullet}%
\parshape 1 \dimexpr\lmargin\relax
    \dimexpr\linewidth-3\rmargin\relax
\lorem

% Third list level
\addtolength\lmargin{30pt}\relax
\hskip-\lmargin\hbox to \lmargin{\hfill\mybullet}%
\parshape 1 \dimexpr\lmargin\relax
    \dimexpr\linewidth-5\rmargin\relax
\lorem

\makeatother
```

- Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par
    - Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par
        - Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

The above were manually defined so that you can understand how the code builds the list. These can be abstracted into a macro similar to **`item`** to typeset the text. The label can be typeset with a new macro **`mklabel`**. We will do this first and change it to an enumerated list. I will use a counter that is already defined for this book and use a macro **`\inc`** to step it up.

Example 2.2 understanding how lists are being build

```
\makeatletter
\def\mybullet{{\itshape\large\inc. }}
\leftskip0pt \rightskip0pt
\setlength\lmargin{0pt}
\setlength\rmargin{0pt}

\def\itm{%
  \setlength\lmargin{0pt}
  \setlength\rmargin{0pt}
  \hskip-\lmargin\hbox to \lmargin{\hfill\mybullet}%
  \parshape 1 \dimexpr\lmargin\relax
      \dimexpr\linewidth-\rmargin\relax}

% First level
\itm \lorem

% Second list level
\itm \lorem

% Third list level
\itm \lorem

\makeatother
```

1 Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par
2 Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par
3 Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

Remember that LaTeX defines all lengths to be global lengths, so we didn't have to redeclare them above, just reset them to the values we needed them.

25.2.1 List and Trivlist

The following variables are used inside a list environment:

?? → P. ?? The distance that the prevailing left margin is indented from the outermost left margin,

?? → P. ?? The width of the current line. Must be initialized to `\hsize`.

?? → P. ?? A count for holding current list nesting depth.

?? → P. ?? 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`.

??[→P.??] 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`.

??[→P.??] When `@inlabel = true`, it holds the labels to be put out by `\everypar`.

??[→P.??] A switch set by `\list` when `@inlabel = true`. Handles the case of a `\list` being the first thing in an item.

??[→P.??] A switch set true for a list that begins an item. No `\topsep` space is added before or after `\item`'s such a list.

??[→P.??] Set true by `\list`, set false by the first text (by `\everypar`).

??[→P.??] Set true when executing an `\item` with no explicit argument. Used to save space. To save time, make two separate **??**[→P.??] commands.

??[→P.??] Set true by `\usecounter` command, causes list to be numbered.

`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 **??**[→P.??] instead of `\hsize`.

Here are the parameters of a list that can be set by commands in the `\list`'s ⟨*commands*⟩ argument. These parameters are all TeX skips or dimensions (defined by `\newskip` or `\newdimen`), so the usual TEX or LATEX commands can be used to set them. The commands will be executed in vmode if and only if the `\list` was preceded by a `\par` (or something like an `\end{list}`), so the spacing parameters can be set according to whether the list is inside a paragraph or is its own paragraph.

Vertical Spacing (skips)

??[→P.??] Space between first item and preceding paragraph.

??[→P.??] Extra space added to **??**[→P.??] when environment starts a new paragraph (is called in vmode).

??[→P.??] Space between successive items.

??[→P.??] Space between paragraphs within an item – the `parskip` for this environment.

Penalties

??[→P.??] put at the beginning of a list

??[→P.??] put at end of list

??[→P.??] put between items.

25.2.2 Horizontal Spacing (dimens)

??[→P.??] space between left margin of enclosing environment (or of page if top level list) and left margin of this list. Must be nonnegative.

??[→P.??] analogous.

??[→P.??] extra indentation at beginning of every paragraph of a list except the one started by the **??**[→P.??] command. May be negative! Usually, labeled lists have **??**[→P.??] equal to zero.

??[→P.??] extra indentation added right BEFORE an item label.

??[→P.??] nominal width of box that contains the label. If the natural width of the label <= **??**[→P.??], then the label is flushed right inside a box of width **??**[→P.??] (with an `\hfil`). Otherwise, a box of the natural width is employed, which causes an indentation of the text on that line.

??[→P.??] space between end of label box and text of first item.

25.2.3 Default Values

Defaults for the list environment are set as follows. First, **??**[→P.??], **??**[→P.??] and **??**[→P.??] are set to opt. Then, one of the commands **??**[→P.??], `@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 **??**[→P.??] to `leftmargini`,..., `leftmarginiv` for the appropriate level. Items that aren't changed may be left alone, but everything that could possibly be changed must be reset.

The **??**[→P.??] simply adjusts the listdepth counter and ends the **??**[→P.??].

The **??**[→P.??] is define as,

The definition of item, is fairly simple deferring the complexity to **??**[→P.??] which follows.

`\topsep`⟨*dim*⟩
The space between the first item and the preceding paragraph.

`\partopsep`⟨*dim*⟩
The space between the first item and the preceding paragraph.

`\itemsep`⟨*dim*⟩
The space between the first item and the preceding paragraph.

`\@itemsep`⟨*dim*⟩
The space between the first item and the preceding paragraph.

`\@topsepadd`⟨*dim*⟩
The space between the first item and the preceding paragraph.

```
??→P.??{⟨LABEL⟩}{⟨COMMANDS⟩} ==
begin
    if ??→P.?? > 5 then
        LaTeX error: 'Too deeply nested'
    else
        ??→P.?? :=G ??→P.?? + 1
    end
    ??→P.?? := opt
    ??→P.?? := opt
    ??→P.?? := opt
    eval(@list ??→P.????→P.????→P.??)
    ??→P.?? :=L LABEL
    ??→P.?? == ??→P.??
    @nmbrlist :=L false
    SETTING COMMANDS
    ??→P.??
    parskip :=L ??→P.??
    parindent :=L ??→P.??
    ??→P.?? :=L ??→P.?? - ??→P.?? -??→P.??
    ??→P.?? :=L ??→P.?? + ??→P.??
    parshape 1 ??→P.?? ??→P.??
    \ignorespaces
end
```

Algorithm 3: The ??→P.?? environment

```
??→P.?? ==
begin
    ??→P.?? :=G ??→P.?? -1
    ??→P.??
end
```

`\outerparskip`⟨*dim*⟩

The space between the first item and the preceding paragraph.

`\parsep`⟨*dim*⟩

The space between the first item and the preceding paragraph.

The skip registers `\topskip`, `\partopsep`, `\itemsep`, `\parsep`, `\@topsep`, `\@topsepadd`, `\outerparskip` are created.

```
\newskip\partopsep
\newskip\itemsep
\newskip\parsep
\newskip\@topsep
\newskip\@topsepadd
\newskip\@outerparskip
```

```
??→P.?? ==
begin
  if @newlist = T then
    ??→P.??
  end
  This command removed for some forgotten reason.
  ??→P.?? :=L ??→P.??
  if @noskipsec then
    leave vertical mode
  end
  if vertical mode then
    ??→P.?? :=L ??→P.?? + ??→P.??
  else
    \unskip \par
  end
  if @inlabel = true then
    @noparitem :=L true @noparlist :=L true
  else
    @noparlist :=L false ??→P.?? :=L ??→P.??
  end
  ??→P.?? :=L ??→P.?? + ??→P.??
  Restore paragraphing parameters
  \leftskip :=L opt
  \rightskip :=L ??→P.??
  \parfillskip :=L 0pt + 1fil
  NOTE: \@setpar called on every ??→P.?? in case \par has been
  temporarily munged before the ??→P.?? command.
  \@setparif @newlist = false then \@@par fi
  \@newlist :=G T
  \@outerparskip :=L \parskip
end
```

```
??→P.?? ==
begin
  ??→P.?? := \parskip
  \@nmbrlist := false
  ??→P.??
  ??→P.?? := 0
  ??→P.?? := 0
  ??→P.?? := ??→P.??
  ??→P.?? :=L "empty"
  ??→P.??{⟨LABEL⟩} == LABEL
end
```

```
??^{→P.??} ==
begin
    if @inlabel = T then
        ??^{→P.??}
    end
    if horizontal mode then
        ??^{→P.??} ??^{→P.??}
    end
    if @noparlist = true then
    else
        if ??^{→P.??} > 0 then
            ??^{→P.??} := ??^{→P.??} ??^{→P.??} - ??^{→P.??} ??^{→P.??} ??^{→P.??}
            -??^{→P.??} + ??^{→P.??}
        end
        ??^{→P.??}
    end
end
```

```
??^{→P.??} == begin
    ??^{→P.??}{⟨@endparpenalty⟩}
    ??^{→P.??}{⟨??^{→P.??}⟩}
    ends the ??^{→P.??} command's ??^{→P.??}
    \endgroup
    \par == begin
        ??^{→P.??}
        \everypar \par
    end
    ??^{→P.??} == begin
        remove ??^{→P.??} ??^{→P.??}
    end
    to match the ??^{→P.??} commands ??^{→P.??}
    ??^{→P.??}
end
```

```
??^{→P.??} == begin
    if math mode then
        issue warning
    end
    if next char = [ then
        ??^{→P.??}
    else
        @noitemarg := true
        ??^{→P.??}[??^{→P.??}]
    end
end
```

Algorithm 4: The algorithm for \item

```
??→P.??[LAB] == begin
  if @noparitem = true then
    @noparitem := false ??→P.????→P.?? :=G
    ??→P.?? begin
      hskip -??→P.??
      box
      refCom@labels
      hskip ??→P.??
    end
    if @minipage = false then
      ??→P.?? := ??→P.??
      ??→P.?? -??→P.??
      ??→P.?? ??→P.?? + ??→P.?? - ??→P.??
    end
  else
    if @inlabel = true then
      then ??→P.?? ??→P.??
    end
    if hmode then
      then 2 ??→P.??'s
      ??→P.??
    end
    if if @newlist = T then
      if @nobreak = T then
        ??→P.??{⟨??→P.?? - ??→P.??⟩}
      else
        ??→P.??{⟨??→P.??⟩}
        ??→P.??{⟨??→P.??⟩}
        ??→P.??{⟨-??→P.??⟩}
      end
    else
      ??→P.??{⟨??→P.??⟩}
      ??→P.??{⟨??→P.??⟩}
    end
    @inlabel :=G true
  end
  ??→P.??{ @minipage :=G F
  @newlist :=G F
  if @inlabel = true then
    @inlabel :=G false ??→P.?? -??→P.??
    ??→P.????→P.??
    ??→P.?? 0
    ??→P.????→P.?? :=G null
  end
  ??→P.??{} }
  @nobreak :=G false
  if @noitemarg = true then
    @noitemarg := false
```



```
      ??→P.??{??→P.??}
    end
  end
  ??→P.?? :=L ??→P.??{⟨??→P.??{⟨LAB⟩}⟩}
  ??→P.????→P.?? :=G ??→P.????→P.????→P.??
  ??→P.?? - (??→P.?? + ??→P.??)
  if ??→P.????→P.?? > ??→P.?? then
    ??→P.????→P.??
```

```
??→P.??{⟨LABEL⟩} == ERROR
default to catch lonely ??→P.??
??→P.??{⟨CTR⟩} == begin
    @nmbrlist :=L true
    @listctr == CTR
    setcounter{⟨CTR⟩}{⟨o⟩}
end
```

\leftmargin⟨*dim*⟩
    \newdimen\leftmargin

\rightmargin⟨*dim*⟩
    \newdimen\rightmargin

\listparindent⟨*dim*⟩
    \newdimen\listparindent

\labelwidth⟨*dim*⟩
    \newdimen\labelwidth

\labelsep⟨*dim*⟩
    \newdimen\labelsep

\linewidth⟨*dim*⟩
    \newdimen\linewidth

\itemindent⟨*dim*⟩
    \newdimen\itemindent

\@totalleftmargin⟨*dim*⟩
    \newdimen\@totalleftmargin \@totalleftmargin=\z@

\leftmargini ⟨*dim*⟩

```
\newdimen\leftmargini
```

\leftmarginii ⟨*dim*⟩

```
\newdimen\leftmarginii
```

\leftmarginiii ⟨*dim*⟩

```
\newdimen\leftmargininiii
```

\leftmarginiv ⟨*dim*⟩

```
\newdimen\leftmarginiv
\newdimen\leftmarginv
```

\@listdepth

\@itempenalty

\@beginparpenalty

\@endparpenalty⟨*penalty*⟩
\newcount\@endparpenalty

```
\newcount\@listdepth \@listdepth=0
\newcount\@itempenalty
\newcount\@beginparpenalty
```

\@labels ⟨*box*⟩

```
\newbox\@labels
```

\if@inlabel
\@inlabeltrue and false

```
\newif\if@inlabel \@inlabelfalse
```

\if@newlist

```
\newif\if@newlist   \@newlistfalse
```

\if@newlistfalse

\if@newlisttrue

\if@noparitem

\if@noparitemtrue

\if@noparitemfalse

```
\newif\if@noparitem \@noparitemfalse
```

\if@noparlist

\if@noparlistfalse

\if@noparlisttrue

```
\newif\if@noparlist \@noparlistfalse
```

\if@noitemarg

\if@noitemargtrue

\if@noitemargfalse

```
\newif\if@noitemarg \@noitemargfalse
```

\if@newlist

\if@newlisttrue

\if@newlistfalse
\@newlist

\if@nmbrlist
Checks if the list is to be numbered or not

```
\newif\if@nmbrlist  \@nmbrlistfalse
```

\@itemlabel
Temporary definition to store the label.

25.3 The \list environment

Notice that the list takes two arguments the first parameter contains instructions as to how to typeset items, that do not have an `\item[]`. If we need to step a counter we tell it here

Example 3.3 The list

```
\@ifundefined{c@steps}{\newcounter{steps}}{}
\begin{list}{\bfseries\upshape Step \arabic{steps}:}
{%
\usecounter{steps}
\setlength{\labelwidth}{2cm}\setlength{\leftmargin}{2.6cm}
\setlength{\labelsep}{0.5cm}\setlength{\rightmargin}{1cm}
\setlength{\parsep}{0.5ex plus0.2ex minus0.1ex}
\setlength{\itemsep}{0ex plus0.2ex minus0pt}\relax
    \slshape %
}
\item Understand how they are being build
\item Define your own environment
\item[Has Own] test
\end{list}


\newenvironment{myplain}
{
  \list{\bfseries\arabic{steps}
      }{\usecounter{steps}\bfseries } }
 {\endlist}


\begin{myplain}
\item \lorem

\item[Something]  \lorem
\end{myplain}
```

{|ifundefinedc@steps

Step 1: *Understand how they are being build*

Step 2: *Define your own environment*

Has Own *test*

1 **Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par**

Something **Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par**

Looking at the `list` environement differntly it is actually a settings environment, it sets what to do if the item has nor argument and also sets the counter to be used the current levels. It end issuing a parshape command that is going to shape the paragraph that follows. Next we need to see how item works in more detail. The calculations and some of the work is delegated to trivlist.

The code follows:

\list{⟨*item label*⟩} {⟨*commands*⟩}

List takes two arguments and is factory command for building other lists.

Check that the list is not nested too far.

```
\def\list#1#2{%
  \ifnum \@listdepth >5\relax
    \@toodeep
  \else
    \global\advance\@listdepth\@ne
  \fi
```

Initialize all margins to opt in case the user did not specify anything.

```
 Example 3.4

  \rightmargin\z@
  \listparindent\z@
  \itemindent\z@
  \csname @list\romannumeral\the\@listdepth\endcsname
   \end{teX}

Define the item label here, if it is empty it will be defined to empty
   \begin{texcode}
   % store the itemlabel
  \def\@itemlabel{#1}%
  \let\makelabel\@mklab
  \@nmbrlistfalse
  % insert other parameters, spaces, fonts etc
   #2\relax
  % call trivlist to do some calculations
  % and set conditionals
  \@trivlist

  % redefine parskip and parindent
  \parskip\parsep
  \parindent\listparindent

  % set the linewidth
  \advance\linewidth -\rightmargin
  \advance\linewidth -\leftmargin
  \advance\@totalleftmargin \leftmargin
```

Finally set the parshape commands. Thsi is perhaps the most misunderstood line in the whole routine. As you see we cannot do any typsetting as yet. Both parameters are just commands to be entered before the text, is typeset. Probably you guessed, but what follows is `\item`. At this point it will just activate an error if it is missing.

```
  \parshape \@ne \@totalleftmargin \linewidth
  \ignorespaces}
```

\par@deathcycles

```
\newcount\par@deathcycles
```

\@trivlist

Auxiliary macro call by trivlist.

Because **\par** is sometimes made a no-op it is possible for a missing **\item** to produce a loop that does not fill memory and so never gets trapped by TEX. We thus need to trap this here by seting **\par** to count the number of times a paragraph ii is called with no progress being made started.

```
\def\@trivlist{%
  \if@noskipsec \leavevmode \fi
  \@topsepadd \topsep
  \ifvmode
    \advance\@topsepadd \partopsep
  \else
    \unskip \par
  \fi
  \if@inlabel
    \@noparitemtrue
    \@noparlisttrue
  \else
    \if@newlist \@noitemerr \fi
    \@noparlistfalse
    \@topsep \@topsepadd
  \fi
  \advance\@topsep \parskip
  \leftskip \z@skip
  \rightskip \@rightskip
  \parfillskip \@flushglue
  \par@deathcycles \z@
  \@setpar{\if@newlist
             \advance\par@deathcycles \@ne
             \ifnum \par@deathcycles >\@m
               \@noitemerr
               {\@@par}%
             \fi
           \else
             {\@@par}%
           \fi}%
  \global \@newlisttrue
  \@outerparskip \parskip}
```

The function of a **trivlist** is very simple, it sets numbering to false and initializes all widths to zero, thus at this stage it will default to a normal paragraph then it calls the auxiliary function **??**$^{\rightarrow \text{P.}??}$, which will set the top and bottom skips and other paragraphing parameters, if set.

Example 3.5 Trivlist

```
\lorem
\begin{trivlist}
```

```
\item[] \lorem

\end{trivlist}
\lorem
```

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

`\trivlist` ⟨*void*⟩

```
\def\trivlist{%
  \parsep\parskip
  \@nmbrlistfalse
  \@trivlist
  \labelwidth\z@
  \leftmargin\z@
  \itemindent\z@
```

We initialise `\@itemlabel` so that a `trivlist` with an `\item` not having an optional argument doesn't produce an error message.

`\makelabel`{⟨*item label*⟩}

```
  \let\@itemlabel\@empty
  \def\makelabel##1{##1}}
```

`\endlist` ⟨*void*⟩

ends the list environment

```
\def\endlist{%
  \global\advance\@listdepth\m@ne
  \endtrivlist}
```

The definition of **??**[→P.??] used to be in ltspace.dtx so that other commands could be 'let to it'. They now use **??**[→P.??].

`\endtrivlist`

```
\def\endtrivlist{%
  \if@inlabel
    \leavevmode
    \global \@inlabelfalse
  \fi
  \if@newlist
    \@noitemerr
    \global \@newlistfalse
  \fi
  \ifhmode\unskip \par
```

We also check if we are in math mode and issue an error message if so (hoping that `\@currenvir` resolves suitably). Otherwise the usual "perhaps a missing item" error will get triggered later which is confusing. [1]

```
  \else
    \@inmatherr{\end{\@currenvir}}%
  \fi
  \if@noparlist \else
    \ifdim\lastskip >\z@
      \@tempskipa\lastskip \vskip -\lastskip
      \advance\@tempskipa\parskip \advance\@tempskipa
            -\@outerparskip
      \vskip\@tempskipa
    \fi
    \@endparenv
  \fi
}
```

\@endparenv

```
\def\@endparenv{%
  \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}
```

\@doendpe

To suppress the paragraph indentation in text immediately following a paragraph-making environment, `\everypar` is changed to remove the space, and `\par` is redefined to restore `\everypar`. Instead of redefining **??**$^{\rightarrow \text{P.}\,??}$ and `\everypar`, **??**$^{\rightarrow \text{P.}\,??}$ was changed to set the **??**$^{\rightarrow \text{P.}\,??}$ switch, letting `end` redefine `par` and `everypar`. This allows paragraph-making environments to work right when called by other environments.)

```
\def\@doendpe{\@endpetrue
     \def\par{\@restorepar\everypar{}
          \par\@endpefalse}\everypar
          {{\setbox\z@\lastbox}\everypar{}\@endpefalse}}
```

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). [2]

[1] v1.0s,2002/10/28,Check for math mode (pr/3437).

[2] v1.0k,1995/11/07,Enclosed **??**$^{\rightarrow \text{P.}\,??}$ assignment by a group so that it leaves the contents of box 0 intact. .

\if@endpe

\@endpelttrue

```
\newif\if@endpe
\@endpefalse
```

\@mklab

```
\def\@mklab#1{\hfil #1}
```

[3] [4] [5]

25.4 Typesetting the label

\item[⟨*label*⟩]

```
\def\item{%
  \@inmatherr\item
  \@ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

\@donoparitem

```
\def\@donoparitem{%
  \@noparitemfalse
  \global\setbox\@labels\hbox{\hskip -\leftmargin
                              \unhbox\@labels
                               \hskip \leftmargin}%
  \if@minipage
    \else
      \@tempskipa\lastskip
      \vskip -\lastskip
      \advance\@tempskipa\@outerparskip
      \advance\@tempskipa -\parskip
      \vskip\@tempskipa
  \fi}
```

\@item{⟨*label*⟩}

```
\def\@item[#1]{%
  \if@noparitem
    \@donoparitem
  \else
    \if@inlabel
      \indent \par
```

[3] LaTeX2.09,1992/09/18,(RmS) Added warning if **??**→P.?? is used in math mode.
[4] v1.0c,1994/04/28,Replaced **??**→P.?? by **??**→P.??.
[5] v1.0d,1994/05/03,Removed superfluous braces.

```
    \fi
    \ifhmode
      \unskip\unskip \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
        \addpenalty\@beginparpenalty
        \addvspace\@topsep
        \addvspace{-\parskip}%
      \fi
    \else
      \addpenalty\@itempenalty
      \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
  \everypar{%
    \@minipagefalse
    \global\@newlistfalse
```

This `\if@inlabel` check is needed in case an item starts of inside a group so that `\everypar` does not become empty outside that group. nobreakfalse, etc etc.

```
    \if@inlabel
      \global\@inlabelfalse
```

The paragraph indent is now removed by using `\setbox...` since this makes `\noindent` a no-op here, as it should be. Thus the following comment is redundant but is left here for the sake of future historians: this next command was changed from an hskip to a kern to avoid a break point after the parindent box: the skip could cause a line-break if a very long label occurs in raggedright setting. [6] [7] [8] If `\noindent` was used after `\item` want to cancel the `\itemindent` skip. This case can be detected as the indentation box will be void.

```
      {\setbox\z@\lastbox
       \ifvoid\z@
         \kern-\itemindent
       \fi}%
```

```
      \box\@labels
      \penalty\z@
    \fi
```

This code is intended to prevent a page break after the first line of an item that comes immediately after a section title. It may be sensible to always forbid a page break after one line of an item? As with all such settings of `\clubpenalty` it is local so will have no effect if the item starts in a group.

Only resetting `\@nobreak` when it is true is now essential since now it is

[6] v1.0d,1994/05/03,**??**→P.**??** changed to **??**→P.**??**.
[7] v1.0m,1996/10/23,**??**→P.**??** changed to **??**→P.**??**.
[8] v1.0r,1997/02/21,**??**→P.**??** check added for **??**→P.**??**. latex/2414.

sometimes set locally. [9]

```
    \if@nobreak
      \@nobreakfalse
      \clubpenalty \@M
    \else
      \clubpenalty \@clubpenalty
      \everypar{}%
    \fi}%
```

[10] [11]

```
  \if@noitemarg
    \@noitemargfalse
    \if@nmbrlist
```

[12]

```
      \refstepcounter\@listctr
    \fi
  \fi
```

Before the label is typeset it is placed in a bo. The latex Team used an **\sbox** so that colour commands can be supported. It call **\makelabel** with the item parameters.

```
  \sbox\@tempboxa{\makelabel{#1}}%
  \global\setbox\@labels\hbox{%
    \unhbox\@labels
    \hskip \itemindent
    \hskip -\labelwidth
    \hskip -\labelsep
    \ifdim \wd\@tempboxa >\labelwidth
       \box\@tempboxa
%
% \changes{LaTeX2.09}{1991/11/22}
%         {(RmS) Changed second call to \refCom{makelabel} to
%           \refCom{unhbox}\refCom{@tempboxa}.
%          Avoids problems with side effects in \refCom{makelabel} and is
%               more efficient.}
%    \begin{teX}
    \else
      \hbox to\labelwidth {\unhbox\@tempboxa}%
    \fix
    \hskip \labelsep}%
  \ignorespaces}
```

\makelabel

Initial definition of **\makelabel** set to produce the infamous and often encountered error, if it has not been defined by one of the paragraphing environments.

[9] v1.0m,1996/10/23,Added setting of **??**→P.?? and set **??**→P.?? only when necessary.

[10] v1.0l,1996/07/26,Remove unecessary **??**→P.?? before **??**→P.??.

[11] v1.0m,1996/10/23,**??**→P.?? moved into the **??**→P.?? and not executed unconditionally, see above.

[12] v1.0g,1995/05/17,Removed surplus braces.

```
\def\makelabel#1{%
  \@latex@error{Lonely \string\item--perhaps a missing
        list environment}\@ehc}
```

\nbitem

[13]

```
\def\@nbitem{%
  \@tempskipa\@outerparskip
  \advance\@tempskipa -\parskip
  \addvspace\@tempskipa}
```

\usecounter

```
\def\usecounter#1{\@nmbrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}
```

25.5 Enumerate

Enumeration is done with four counters: **enumi**, **enumii**, **enumiii** and **enumiv**, where **enum**N controls the numbering of the Nth level enumeration. The label is generated by the commands **??**[→P.??] ... **\labelenumiv**, which should be defined by the document style.

Note that **\p@enum**N **\theenum**N defines the output of a **??**[→P.??] 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 **??**[→P.??] as '3a'.

The item numbers are moved to the right of the label box, so they are always a distance of **??**[→P.??] from the item.

??[→P.??] holds the current enumeration nesting depth.

Itemization is controlled by four commands: **\labelitemi**, **\labelitemii**, **\labelitemiii**, and **\labelitemiv**. To cause the second-level list to be bulleted, you just define **\labelitemii** to be **$\bullet$**. **@itemspacing** and **??**[→P.??] are the analogs of **@enumspacing** and **??**[→P.??]. These are not defined in the kernel, their definition is delegated to the classes and to another extend to the users.

\@enumdepth

The enumeration depth.

[13] v1.0g,1995/05/17,Removed surplus braces.

```
\newcount\@enumdepth \@enumdepth = 0
```

Next the counters for the `\env:enumerate`environment, `c@enumi`, `c@enumii`, `c@enumiii` and `c@enumiv` are created.

```
 \@definecounter{enumi}
 \@definecounter{enumii}
 \@definecounter{enumiii}
 \@definecounter{enumiv}
```

`\begin{enumerate}`
⟨*environment content*⟩
`\end{enumerate}`

The enumerate environment enumerates the list. The macro is written very efficiently and defines the basic structure. The typesetting parameters are left for the class files such as book to define them.

```
 \def\enumerate{%
  \ifnum \@enumdepth >\thr@@\@toodeep\else
    \advance\@enumdepth\@ne
    \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
      \expandafter
      \list
         \csname label\@enumctr\endcsname
        {\usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
  \fi}
 \let\endenumerate =\endlist
```

`\@itemdepth`

```
\newcount\@itemdepth \@itemdepth = 0
```

25.6 The itemize environment

Finally the `itemize` environment is defined. This is defined in the kernel to have a limit of 4 levels only.

Itemization is controlled by four commands, which are defined in the standard classes: `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`, which define the labels of the various itemization levels: the symbols used in the standard classes are bullet, bold en-dash, centered asterisk and centred dot.

`\begin{itemize}`
⟨*environment content*⟩
`\end{itemize}`

```
 \def\itemize{%
  \ifnum \@itemdepth >\thr@@\@toodeep\else
    \advance\@itemdepth\@ne
    \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}% ❶
```

The ❶ is a hack to create the `\@itemi`...`\@itemv` names. How it works is that the `\romannumeral` will expand until it finds a number and leave the letter `i` etc. in the stream.

Example 6.6 Creating a suffix with Roman Numerals

```
\newcounter{tempcounter}
\setcounter{tempcounter}{5}
\edef\itemtest{labelitem\romannumeral\thetempcounter}
%  the replacement
\meaning\itemtest

\itemtest
```

macro:->labelitemv
labelitemv

A romannumeral just returns nothing if it encounters the number `0` or a negative number.

Example 6.7 Creating a suffix with Roman Numerals

```
% counter has been created globally
% commented out.
%\newcounter{tempcounter}
\setcounter{tempcounter}{0}
\edef\itemtest{labelitem\romannumeral\thetempcounter}
%  the replacement
\meaning\itemtest\par
\itemtest
```

macro:->labelitem
labelitem

Call the **??**→P.?? sets all parameters and defines a **??**→P.?? for this particular environment. This is a good place to also hook other items to decorate the label, if necessary. Remember also that the `\@itemitem`, defined above is just there to facilitate calling `labelitemi` ...`labelitemn` depending on the level.

```
    \expandafter
    \list

      \csname\@itemitem\endcsname
      {\def\makelabel##1{\hss\llap{##1}}}%
\fi}
```

The astude reader will notice at this point that although we defined the name of labeli etc, they still have no meaning. As mentioned earlier there is still some work to be done before we have full lists. These are defined in the standatrd classes and the definitions are shown below.

revised
13-06-2018

```
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}
```

Finally the end of the environment is defined.

```
\let\enditemize =\endlist
```

phd package and give a background as to how fonts are used in LaTeX.

We will make our products work out of the box.

Steve Balmer

26.1 Typesetting boxed content

Boxes are an important part of TeX's typography and LaTeX provides numerous macros for handling these, both at the user level, as well as the kernel level. This section of the kernel deals with both storing material in a box to be typeset at a later stage, as well as providing commands for typesetting text within framed or unframed boxes. R 13/01/2018

Many packages exist that modify these commands and providing settings to typeset a bewilding variety of boxed content. Perhaps the best one so far is **tcolorbox**.

??[→ P. **??**][⟨*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}}`

Example 1.1 makebox

```
\bgroup
\parindent0pt
\makebox[2\width][l]{example box}\\
\makebox[2\width][c]{example box}\\
\makebox[2\width][r]{example box}\\
\framebox[2\width][s]{example box}
\egroup
```

example box
example box
example box

example box

??[→ P. ??] `\underline{⟨text⟩}` : Makes an underlined `\hbox` with ⟨*text*⟩ in it.

```
%<*2ekernel>
\message{boxes,}
```

\makebox

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\makebox}{Make \makebox ↲
      ↳robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\makebox{%
  \leavevmode
  \@ifnextchar(%)
    \@makepicbox
    {\@ifnextchar[\@makebox\mbox}}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\makebox}{Make \makebox ↲
      ↳robust}%
%<latexrelease>\def\makebox{%
%<latexrelease>  \leavevmode
%<latexrelease>  \@ifnextchar(%)
%<latexrelease>    \@makepicbox
%<latexrelease>    {\@ifnextchar[\@makebox\mbox}}%
%<latexrelease>\expandafter\let\csname makebox ↲
      ↳\endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

\mbox{⟨*material*⟩}

The basic horizontal box command for LaTeX. This starts a paragraph and puts its contents in an `\hbox`. As expected it will not wrap its text.

```
\long\def\mbox#1{\leavevmode\hbox{#1}}
```

Example 1.2 mbox

```
\mbox{\fox}
```

The quick brown fox jumps over the lazy dogs.

`\@makebox` Look for a possible second optional argument (defaults to `c`).

```
\def\@makebox[#1]{%
  \@ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}
```

(*End definition for* `\@makebox`.)

`\@begin@tempboxa` Helper macro for supporting **\height**, **\width** etc. Grab **#1** into **\@tempboxa** and measure it.

```
\long\def\@begin@tempboxa#1#2{%
   \begingroup
     \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
     \def\width{\wd\@tempboxa}%
     \def\height{\ht\@tempboxa}%
     \def\depth{\dp\@tempboxa}%
     \let\totalheight\@ovri
     \totalheight\height
     \advance\totalheight\depth}
```

(*End definition for* `\@begin@tempboxa`.)

`\@end@tempboxa` End the group started by **\@begin@tempboxa**, so that the scope of **\height** only includes the 'length' argument to the user-command.

```
\let\@end@tempboxa\endgroup
```

(*End definition for* `\@end@tempboxa`.)

`\bm@c` `\bm@l` `\bm@r` `\bm@s` `\bm@t` `\bm@b` Set up spacing.

```
\def\bm@c{\hss\unhbox\@tempboxa\hss}
\def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
\def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
\def\bm@s{\unhbox\@tempboxa}
```

(*End definition for* `\bm@c` *and others*.)

`\@imakebox`

```
\long\def\@imakebox[#1][#2]#3{%
  \@begin@tempboxa\hbox{#3}%
    \setlength\@tempdima{#1}%       support calc
    \hb@xt@\@tempdima{\csname bm@#2\endcsname}%
  \@end@tempboxa}
```

(*End definition for* `\@imakebox`.)

`\@makepicbox` Picture mode form of **\makebox**.

```
\def\@makepicbox(#1,#2){%
  \@ifnextchar[{\@imakepicbox(#1,#2)}{\@imakepicbox(#1,#2)[]}}
```

(*End definition for* `\@makepicbox`.)

`\@imakepicbox` picture mode version

```
\long\def\@imakepicbox(#1,#2)[#3]#4{%
  \vbox to#2\unitlength
   {\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
    \let\mb@t\vss
    \@tfor\reserved@a :=#3\do{%
      \if s\reserved@a
        \let\mb@l\relax\let\mb@r\relax
      \else
        \expandafter\let\csname mb@\reserved@a\endcsname\relax
      \fi}%
    \mb@t
    \hb@xt@ #1\unitlength{\mb@l #4\mb@r}%
    \mb@b
```

This kern ensures that a **b** option aligns on the bottom of the text rather than the baseline. this is the documented behaviour in the LaTeXBook. The kern is removed in compatibility mode.

```
    \kern\z@}}
```

(*End definition for* `\@imakepicbox`.)

`\set@color`

This macro is initially a no-op, but the colour package will redefine it to insert a `\special`.

```
\let\set@color\relax
```

(*End definition for* `\set@color`.)

`\color@begingroup` `\color@endgroup` `\color@setgroup` `\normalcolor` `\color@hbox` `\color@vbox` `\color@endbox`

These macros are initially a no-op, but the colour package will redefine them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`, `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and ⟨*set to main document colour*⟩ respectively.

```
\let\color@begingroup\relax
\let\color@endgroup\relax
\let\color@setgroup\relax
\let\normalcolor\relax
\let\color@hbox\relax
\let\color@vbox\relax
\let\color@endbox\relax
```

(*End definition for* `\color@begingroup` *and others.*)

26.2 Saving material in a box

To save material in a variable we can use the **??**[→ P. ??] to create the variable and the **??**[→ P. ??] to save it.

`\newsavebox`

Allocate a new 'savebox'.

```
\def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}
```

\savebox

Save **#1** in a box register.

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\savebox}{Make \savebox ⤸
      ⤹robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\savebox[1]{%
  \@ifnextchar(%)
    {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\savebox}{Make \savebox ⤸
      ⤹robust}%
%<latexrelease>\def\savebox#1{%
%<latexrelease>  \@ifnextchar(%)
%<latexrelease>    ⤸
      ⤹{\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
%<latexrelease>\expandafter\let\csname savebox ⤸
      ⤹\endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

\sbox{⟨*box register*⟩}

Save **#1** in a box register.

```
\long\def\sbox#1#2{\setbox#1\hbox{%
  \color@setgroup#2\color@endgroup}}
```

```
\def\@savebox#1[#2]{%
  \@ifnextchar [{\@isavebox#1[#2]}{\@isavebox#1[#2][c]}}
```

```
\long\def\@isavebox#1[#2][#3]#4{%
  \sbox#1{\@imakebox[#2][#3]{#4}}}
```

```
\def\@savepicbox#1(#2,#3){%
  \@ifnextchar[%]
    {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3)[]}}
```

```
\long\def\@isavepicbox#1(#2,#3)[#4]#5{%
  \sbox#1{\@imakepicbox(#2,#3)[#4]{#5}}}
```

\begin{**lrbox**}{⟨*material*⟩}⟨*environment content*⟩\end{**lrbox**}

lrbox: the new environment form of **\sbox**. Use **\aftergroup** tricks to enable a *local* assignment to be made to the box, in a way that it still has an effect *outside* the **lrbox** environment.

```
\def\lrbox#1{%
  \edef\reserved@a{%
    \endgroup
    \setbox#1\hbox{%
```

```
      \begingroup\aftergroup}%
        \def\noexpand\@currenvir{\@currenvir}%
        \def\noexpand\@currenvline{\on@line}}%
  \reserved@a
    \@endpefalse
    \color@setgroup
      \ignorespaces}
```

\endlrbox

```
\def\endlrbox{\unskip\color@endgroup}
```

(*End definition for* `\endlrbox`.)

\usebox{⟨*name*⟩}

Convenience command to typeset the box contents. It uses copy so that it does not destroy the contents of the **\box**.

```
\def\usebox#1{\leavevmode\copy #1\relax}
```

\frame

The following definition of **\frame** was written by Pavel Curtis

```
\long\def\frame#1{%
  \leavevmode
  \hbox{%
    \hskip-\@wholewidth
    \vbox{%
      \vskip-\@wholewidth
      \hrule \@height\@wholewidth
      \hbox{%
        \vrule\@width\@wholewidth
        #1%
        \vrule\@width\@wholewidth}%
      \hrule\@height\@wholewidth
      \vskip-\@wholewidth}%
    \hskip-\@wholewidth}}
```

(*End definition for* `\frame`.)

\fboxrule

User macro to set the width of the **fbox** rule.

\fboxsep

User level macro to set the padding of an **\fbox.**

```
\newdimen\fboxrule
\newdimen\fboxsep
```

\fbox

```
\long\def\fbox#1{%
  \leavevmode
  \setbox\@tempboxa\hbox{%
    \color@begingroup
      \kern\fboxsep{#1}\kern\fboxsep
    \color@endgroup}%
  \@frameb@x\relax}
```

\framebox{⟨*material*⟩}

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\framebox}{Make \framebox ⤶
      ⤷robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\framebox{%
  \@ifnextchar(%)
    \@framepicbox{\@ifnextchar[\@framebox\fbox}}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\framebox}{Make \framebox ⤶
      ⤷robust}%
%<latexrelease>\def\framebox{%
%<latexrelease>  \@ifnextchar(%)
%<latexrelease>    ⤶
      ⤷\@framepicbox{\@ifnextchar[\@framebox\fbox}}%
%<latexrelease>\expandafter\let\csname framebox ⤶
      ⤷\endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

\@framebox Deal with optional arguments.

```
\def\@framebox[#1]{%
  \@ifnextchar[%]
    {\@iframebox[#1]}%
    {\@iframebox[#1][c]}}
```

(*End definition for* `\@framebox`.)

\@iframebox The handling the optional arguments. In order to set the whole box, including the frame to the specified dimension, we first determine that dimension from the natural size of the text, `#3`. calculated width.

```
\long\def\@iframebox[#1][#2]#3{%
  \leavevmode
  \@begin@tempboxa\hbox{#3}%
    \setlength\@tempdima{#1}%
    \setbox\@tempboxa\hb@xt@\@tempdima
         {\kern\fboxsep\csname bm@#2\endcsname\kern\fboxsep}%
    \@frameb@x{\kern-\fboxrule}%
  \@end@tempboxa}
```

(*End definition for* `\@iframebox`.)

```
\def\@frameb@x#1{%
  \@tempdima\fboxrule
  \advance\@tempdima\fboxsep
  \advance\@tempdima\dp\@tempboxa
  \hbox{%
    \lower\@tempdima\hbox{%
      \vbox{%
        \hrule\@height\fboxrule
        \hbox{%
          \vrule\@width\fboxrule
          #1%
          \vbox{%
            \vskip\fboxsep
            \box\@tempboxa
            \vskip\fboxsep}%
          #1%
          \vrule\@width\fboxrule}%
        \hrule\@height\fboxrule}%
                            }%
        }%
}
\def\@framepicbox(#1,#2){%
  \@ifnextchar[{\@iframepicbox(#1,#2)}{\@iframepicbox(#1,#2)[]}}
\long\def\@iframepicbox(#1,#2)[#3]#4{%
  \frame{\@imakepicbox(#1,#2)[#3]{#4}}}
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\parbox}{Make \parbox robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\parbox{%
  \@ifnextchar[%]
    \@iparbox
    {\@iiiparbox c\relax[s]}}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\parbox}{Make \parbox robust}%
%<latexrelease>\def\parbox{%
%<latexrelease>  \@ifnextchar[%]
%<latexrelease>    \@iparbox
%<latexrelease>    {\@iiiparbox c\relax[s]}}%
%<latexrelease>\expandafter\let\csname parbox ↩
      ↪\endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
\def\@iparbox[#1]{%
  \@ifnextchar[%]
    {\@iiparbox{#1}}%
    {\@iiiparbox{#1}\relax[s]}}
\def\@iiparbox#1[#2]{%
  \@ifnextchar[%]
    {\@iiiparbox{#1}{#2}}%
```

```
    {\@iiiparbox{#1}{#2}[#1]}}
\let\@parboxto\@empty
\long\def\@iiiparbox#1#2[#3]#4#5{%
  \leavevmode
  \@pboxswfalse
  \setlength\@tempdima{#4}%
  \@begin@tempboxa\vbox{\hsize\@tempdima\@parboxrestore#5\@@par}%
    \ifx\relax#2\else
      \setlength\@tempdimb{#2}%
      \edef\@parboxto{to\the\@tempdimb}%
    \fi
    \if#1b\vbox
    \else\if #1t\vtop
    \else\ifmmode\vcenter
    \else\@pboxswtrue $\vcenter
    \fi\fi\fi
    \@parboxto{\let\hss\vss\let\unhbox\unvbox
       \csname bm@#3\endcsname}%
    \if@pboxsw \m@th$\fi
  \@end@tempboxa}
%</2ekernel>
%<latexrelease>\IncludeInRelease{2017-04-15}%
%<latexrelease>                 {\normallineskiplimit}
%<latexrelease>                 {reset \lineskiplimit}%
%<*2ekernel|latexrelease>
\def\@arrayparboxrestore{%
  \let\if@nobreak\iffalse
  \let\if@noskipsec\iffalse
  \let\par\@@par
  \let\-\@dischyph
  \let\'\@acci\let\`\@accii\let\=\@acciii
  \parindent\z@ \parskip\z@skip
  \everypar{}%
  \linewidth\hsize
  \@totalleftmargin\z@
  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
  \parfillskip\@flushglue
  \lineskip\normallineskip
  \lineskiplimit\normallineskiplimit
  \baselineskip\normalbaselineskip
  \sloppy}
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000-00-00}%
%<latexrelease>                 {\normallineskiplimit}
%<latexrelease>                 {reset \lineskiplimit}%
%<latexrelease>\def\@arrayparboxrestore{%
%<latexrelease>  \let\if@nobreak\iffalse
%<latexrelease>  \let\if@noskipsec\iffalse
%<latexrelease>  \let\par\@@par
%<latexrelease>  \let\-\@dischyph
```

```
%<latexrelease>  \let\'\@acci\let\`\@accii\let\=\@acciii
%<latexrelease>  \parindent\z@ \parskip\z@skip
%<latexrelease>  \everypar{}%
%<latexrelease>  \linewidth\hsize
%<latexrelease>  \@totalleftmargin\z@
%<latexrelease>  \leftskip\z@skip \rightskip\z@skip ↩
      ↪\@rightskip\z@skip
%<latexrelease>  \parfillskip\@flushglue ↩
      ↪\lineskip\normallineskip
%<latexrelease>  \baselineskip\normalbaselineskip
%<latexrelease>  \sloppy}
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

```
\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
```

`\if@minipage` Switch that is true at the start of a minipage.

```
\def\@minipagefalse{\global\let\if@minipage\iffalse}
\def\@minipagetrue {\global\let\if@minipage\iftrue}
\@minipagefalse
```

(*End definition for* `\if@minipage`.)

`\minipage` Essentially an environment form of `\parbox`.

```
\def\minipage{%
  \@ifnextchar[%]
    \@iminipage
    {\@iiiminipage c\relax[s]}}
```

(*End definition for* `\minipage`.)

`\@iminipage` Optional argument handling.

```
\def\@iminipage[#1]{%
  \@ifnextchar[%]
    {\@iiminipage{#1}}%
    {\@iiiminipage{#1}\relax[s]}}
```

(*End definition for* `\@iminipage`.)

`\@iiminipage` Optional argument handling.

```
\def\@iiminipage#1[#2]{%
  \@ifnextchar[%]
    {\@iiiminipage{#1}{#2}}%
    {\@iiiminipage{#1}{#2}[#1]}}
```

(*End definition for* `\@iiminipage`.)

`\@iiiminipage`

```
\def\@iiiminipage#1#2[#3]#4{%
  \leavevmode
  \@pboxswfalse
```

```
  \setlength\@tempdima{#4}%
  \def\@mpargs{{#1}{#2}[#3]{#4}}%
  \setbox\@tempboxa\vbox\bgroup
    \color@begingroup
      \hsize\@tempdima
      \textwidth\hsize \columnwidth\hsize
      \@parboxrestore
      \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
      \let\@footnotetext\@mpfootnotetext
      \let\@listdepth\@mplistdepth \@mplistdepth\z@
      \@minipagerestore
      \@setminipage}
```

(*End definition for* `\@iiiminipage`.)

`\@minipagerestore` Hook so that other styles can reset other commands in a minipage.

```
 \let\@minipagerestore=\relax
```

(*End definition for* `\@minipagerestore`.)

\endminipage

```
\def\endminipage{%
    \par
    \unskip
    \ifvoid\@mpfootins\else
      \vskip\skip\@mpfootins
      \normalcolor
      \footnoterule
      \unvbox\@mpfootins
    \fi
    \@minipagefalse   %% added 24 May 89
  \color@endgroup
  \egroup
  \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
```

`\@mplistdepth` `\@mpfootins`

```
\newcount\@mplistdepth
\newinsert\@mpfootins
```

(*End definition for* `\@mplistdepth` *and* `\@mpfootins`.)

```
\long\def\@mpfootnotetext#1{%
  \global\setbox\@mpfootins\vbox{%
    \unvbox\@mpfootins
    \reset@font\footnotesize
    \hsize\columnwidth
    \@parboxrestore
    \protected@edef\@currentlabel
         {\csname p@mpfootnote\endcsname\@thefnmark}%
    \color@begingroup
      \@makefntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
    \color@endgroup}}
```

```
\newif\if@pboxsw
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\rule}{Make \rule robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\rule}{Make \rule robust}%
%<latexrelease>\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
%<latexrelease>\expandafter\let\csname rule \endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
 \def\@rule[#1]#2#3{%
   \leavevmode
   \hbox{%
     \setlength\@tempdima{#1}%
     \setlength\@tempdimb{#2}%
     \setlength\@tempdimc{#3}%
     \advance\@tempdimc\@tempdima
     \vrule\@width\@tempdimb\@height\@tempdimc\@depth-\@tempdima}}
\let\@@underline\underline
\def\underline#1{%
  \relax
  \ifmmode\@@underline{#1}%
  \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}
%</2ekernel>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\raisebox}{Make \raisebox \robust}%
%<*2ekernel|latexrelease>
\DeclareRobustCommand\raisebox[1]{%
  \leavevmode
  \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\raisebox}{Make \raisebox \robust}%
%<latexrelease>\def\raisebox#1{%
%<latexrelease>  \leavevmode
%<latexrelease>  \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
%<latexrelease>\expandafter\let\csname raisebox \endcsname\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
\def\@rsbox#1[#2]{%
  \@ifnextchar[{\@iirsbox{#1}[#2]}{\@irsbox{#1}[#2]}}
```

```
\long\def\@irsbox#1[#2]#3{%
  \@begin@tempboxa\hbox{#3}%
    \setlength\@tempdima{#1}%
    \ifx\\#2\\\else\setlength\@tempdimb{#2}\fi
    \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
    \ifx\\#2\\\else\ht\@tempboxa\@tempdimb\fi
    \box\@tempboxa
  \@end@tempboxa}
```

```
\long\def\@iirsbox#1[#2][#3]#4{%
  \@begin@tempboxa\hbox{#4}%
    \setlength\@tempdima{#1}%
    \setlength\@tempdimb{#2}%
    \setlength\dimen@{#3}%
    \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
    \ht\@tempboxa\@tempdimb
    \dp\@tempboxa\dimen@
    \box\@tempboxa
  \@end@tempboxa}
```

`\@finalstrut` This macro adds a special strut the *depth* of the box given as `#1`, and height and width 0pt. It is used for ensuring that the last line of a paragraph has the correct depth in 'p' columns of tables and in footnotes. In vertical mode nothing is done, as adding the strut (as done in 2.09) would start a new paragraph. It would be possible to inspect `\prevdepth` to check the depth of the just-completed paragraph, but we do not do that here. The `\nobreak` was added (1995/10/31) to allow hyphenation of the final word of the paragraph.

```
\def\@finalstrut#1{%
  \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}
```

(*End definition for* `\@finalstrut`.)

26.2.1 Some low-level constructs

The following commands are basically inherited from plain TeX. Note the us of `\@@line` to refer to the original Plain TeX macros.

`\leftline`{⟨*material*⟩}

```
\def\leftline#1{\@@line{#1\hss}}
```

`\rightline` `\centerline` `\@@line` These macros place text on a full line either centred or left or right adjusted.

```
\def\@@line{\hb@xt@\hsize}
\def\rightline#1{\@@line{\hss#1}}
\def\centerline#1{\@@line{\hss#1\hss}}
```

(*End definition for* `\rightline`, `\centerline`, *and* `\@@line`.)

Finally the lapping macros are defined.

\rlap
\llap

These macros place text to the left or right of the current reference point without taking up space.

```
\def\rlap#1{\hb@xt@\z@{#1\hss}}
\def\llap#1{\hb@xt@\z@{\hss#1}}
```

(*End definition for* `\rlap` *and* `\llap`.)

26.3 Concluding remarks

The macros just defined are really core concepts introduced in LaTeX 2e. Many package authors have produced similar macros to improve the looks of the boxing commands. The LaTeX 3 Team is currently working to produce some of these concepts as part of the **xpackages**. However many of what has been described here will probbaly substantially change at the author level, through the use of templates.

```
%</2ekernel>
```

phd package and give a background as to how fonts are used in LaTeX.

This section deals with 'Lining It Up in Columns'. First the **tabbing** environment is defined, and then in second part, **tabular** together with its variants, **tabular*** and **array**. Note that the **tabular** defined here is essentially the original LaTeX 2.09 version, not the extended version described in *The LaTeX Companion*. Use the **array** package to obtain the extended version.

27.0.1 tabbing

\a The accents \` , \' , and \= that have been redefined inside a tabbing environment can be called by typing \a` , \a' , and \a=. The macro \a is defined in **ltoutenc.dtx**.

(*End definition for* \a.)

```
%<2ekernel>\expandafter\let\csname ⤸
      ⤹ver@autotabg.sty\endcsname\fmtversion
%<*2ekernel>
\newdimen\@gtempa
\chardef\@firsttab=\the\allocationnumber
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen ⤸
      ⤹\@gtempa
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen ⤸
      ⤹\@gtempa
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen ⤸
      ⤹\@gtempa
\newdimen\@gtempa
\chardef\@maxtab=\the\allocationnumber
\dimen\@firsttab=0pt
\newcount\@nxttabmar
\newcount\@curtabmar
\newcount\@curtab
\newcount\@hightab
\newcount\@tabpush
\newbox\@curline
\newbox\@curfield
\newbox\@tabfbox
\newif\if@rjfield
\gdef\@startline{%
     \ifnum \@nxttabmar >\@hightab
       \@badtab
       \global\@nxttabmar \@hightab
     \fi
     \global\@curtabmar \@nxttabmar
     \global\@curtab \@curtabmar
     \global\setbox\@curline \hbox {}%
     \@startfield
     \strut}
\gdef\@stopline{%
  \unskip
  \@stopfield
```

```
  \if@rjfield
    \global\@rjfieldfalse
    \@tempdima\@totalleftmargin
    \advance\@tempdima\linewidth
    \hb@xt@\@tempdima{%
      \@itemfudge\hskip\dimen\@curtabmar
      \box\@curline
      \hfil
      \box\@curfield}%
  \else
    \@addfield
    \hbox{\@itemfudge\hskip\dimen\@curtabmar\box\@curline}%
  \fi}
\gdef\@startfield{%
  \global\setbox\@curfield\hbox\bgroup\color@begingroup}
\gdef\@stopfield{%
  \color@endgroup\egroup}
\gdef\@contfield{%
  \global\setbox\@curfield\hbox\bgroup\color@begingroup
  \unhbox\@curfield}
\gdef\@addfield{\global\setbox\@curline\hbox{\unhbox
      \@curline\unhbox\@curfield}}
\gdef\@ifatmargin{\ifdim \wd\@curline =\z@}
\gdef\@tabcr{\@stopline \@ifstar{\penalty \@M ⤸
      ⤹\@xtabcr}\@xtabcr}
\gdef\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}
\gdef\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
\gdef\kill{\@stopfield\@startline\ignorespaces}
\gdef\tabbing{\lineskip ⤸
      ⤹\z@skip\let\>\@rtab\let\<\@ltab\let\=\@settab
      \let\+\@tabplus\let\-\@tabminus\let\`\@tabrj\let\'\@tablab
      \let\\=\@tabcr
      \@hightab\@firsttab
      \global\@nxttabmar\@firsttab
      \dimen\@firsttab\@totalleftmargin
      \global\@tabpush\z@ \global\@rjfieldfalse
      \trivlist \item\relax
      \if@minipage\else\vskip\parskip\fi
      \setbox\@tabfbox\hbox{%
        \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
      \def\@itemfudge{\box\@tabfbox}%
      \@startline\ignorespaces}
\gdef\endtabbing{%
  \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}
\gdef\@rtab{\@stopfield\@addfield\ifnum \@curtab<\@hightab
       \global\advance\@curtab \@ne \else\@badtab\fi
       \@tempdima\dimen\@curtab
```

```
      \advance\@tempdima -\dimen\@curtabmar
      \advance\@tempdima -\wd\@curline
      \global\setbox\@curline\hbox{\unhbox\@curline\hskip\@tempdima}%
      \@startfield\ignorespaces}
\gdef\@settab{\@stopfield\@addfield
  \ifnum \@curtab <\@maxtab
    \ifnum\@curtab =\@hightab
      \advance\@hightab \@ne
    \fi
    \global\advance\@curtab \@ne
  \else
    \@latex@error{Tab overflow}\@ehd
  \fi
  \dimen\@curtab \dimen\@curtabmar
  \advance\dimen\@curtab \wd\@curline
  \@startfield
  \ignorespaces}
\gdef\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firsttab
      \global\advance\@curtab \m@ne ⤸
            ⤹\global\advance\@curtabmar\m@ne\else
      \@badtab\fi\else
      \@latex@error{\string\<\space in mid ⤸
            ⤹line}\@ehd\fi\ignorespaces}
\gdef\@tabplus{%
  \ifnum\@nxttabmar<\@hightab
    \global\advance\@nxttabmar\@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
\gdef\@tabminus{%
  \ifnum\@nxttabmar>\@firsttab
    \global\advance\@nxttabmar\m@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
\gdef\@tabrj{%
  \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}
\gdef\@tablab{%
  \@stopfield
  \global\setbox\@curline\hbox{%
    \box\@curline
    \hskip-\wd\@curfield \hskip-\tabbingsep
    \box\@curfield
    \hskip\tabbingsep}%
  \@startfield
  \ignorespaces}
\gdef\pushtabs{%
  \@stopfield\@addfield\global\advance\@tabpush \@ne ⤸
        ⤹\begingroup
```

```
      \@contfield}
\gdef\poptabs{\@stopfield\@addfield
  \ifnum \@tabpush >\z@
    \endgroup
    \global\advance\@tabpush \m@ne
    \ifnum \@curtab >\@hightab
      \global \@curtab \@hightab
      \@badtab
    \fi
  \else
    \@badpoptabs
  \fi
  \@contfield}
\newdimen\tabbingsep
```

27.1 array and tabular environments

```
\def\extracolsep#1{\tabskip #1\relax}
\def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
 \let\@classiv\@arrayclassiv
 \let\\\@arraycr\let\@halignto\@empty\@tabarray}
\def\endarray{\crcr\egroup\egroup}
\def\endtabular{\crcr\egroup\egroup $\egroup}
\expandafter \let \csname endtabular*\endcsname = \endtabular
\def\tabular{\let\@halignto\@empty\@tabular}
\@namedef{tabular*}#1{%
 \setlength\dimen@{#1}%
   \edef\@halignto{to\the\dimen@}\@tabular}
\def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
   \let\@classz\@tabclassz
   \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}
\def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}
\def\@array[#1]#2{%
  \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
  \bgroup
  \setbox\@arstrutbox\hbox{%
    \vrule \@height\arraystretch\ht\strutbox
           \@depth\arraystretch \dp\strutbox
           \@width\z@}%
  \@mkpream{#2}%
  \edef\@preamble{%
    \ialign \noexpand\@halignto
      \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%
  \let\@startpbox\@@startpbox \let\@endpbox\@@endpbox
  \let\tabularnewline\\%
    \let\par\@empty
```

```
    \let\@sharp##%
    \set@typeset@protect
    \lineskip\z@skip\baselineskip\z@skip
    \ifhmode \@preamerr\z@ \@@par\fi
    \@preamble}
\def\@arraycr{%
  ${\ifnum0=`}\fi\@ifstar\@xarraycr\@xarraycr}
\def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0=`{\fi}${}\cr}}
\def\@argarraycr[#1]{%
  \ifnum0=`{\fi}${}\ifdim #1>\z@ \@xargarraycr{#1}\else
   \@yargarraycr{#1}\fi}
\let\tabularnewline\relax
\def\@tabularcr{%
  {\ifnum0=`}\fi\@ifstar\@xtabularcr\@xtabularcr}
\def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0=`{\fi}\cr}}
\def\@argtabularcr[#1]{%
  \ifnum0=`{\fi}%
    \ifdim #1>\z@
      \unskip\@xargarraycr{#1}%
    \else
      \@yargarraycr{#1}%
    \fi}
\def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
   \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}
\def\@yargarraycr#1{\cr\noalign{\vskip #1}}
\long\def\multicolumn#1#2#3{\multispan{#1}\begingroup
  \@mkpream{#2}%
  \def\@sharp{#3}\set@typeset@protect
  \let\@startpbox\@@startpbox\let\@endpbox\@@endpbox
  \@arstrut \@preamble\hbox{}\endgroup\ignorespaces}
\def\@xexnoop #1\@@{}
\def\@expast#1{\@xexpast #1*0x\@@}
\def\@xexpast#1*#2#3#4\@@{%
  \edef\reserved@a{#1}%
  \@tempcnta#2\relax
  \ifnum\@tempcnta>\z@
    \@whilenum\@tempcnta>\z@\do
       {\edef\reserved@a{\reserved@a#3}\advance\@tempcnta \m@ne}%
    \let\reserved@b\@xexpast
  \else
    \let\reserved@b\@xexnoop
  \fi
  \expandafter\reserved@b\reserved@a #4\@@}
\newif\if@firstamp
```

```
\def\@addamp{%
  \if@firstamp
    \@firstampfalse
  \else
    \edef\@preamble{\@preamble &}%
  \fi}
\def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\def\@ampacol{\@addamp \@acol}
\def\@acolampacol{\@acol\@addamp\@acol}
\def\@mkpream#1{\@firstamptrue\@lastchclass6
  \let\@preamble\@empty
  \let\protect\@unexpandable@protect
  \let\@sharp\relax
  \let\@startpbox\relax\let\@endpbox\relax
  \@expast{#1}%
  \expandafter\@tfor \expandafter
    \@nextchar \expandafter:\expandafter=\reserved@a\do
       {\@testpach\@nextchar
    \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
      \or \@classiv \or\@classv \fi\@lastchclass\@chclass}%
  \ifcase \@lastchclass \@acol
      \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi}
\def\@arrayclassz{\ifcase \@lastchclass \@acolampacol \or \@ampacol \or
   \or \or \@addamp \or
   \@acolampacol \or \@firstampfalse \@acol \fi
\edef\@preamble{\@preamble
  \ifcase \@chnum
     \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
    \or \hfil$\relax\@sharp$\fi}}
\def\@tabclassz{%
  \ifcase\@lastchclass
    \@acolampacol
  \or
    \@ampacol
  \or
  \or
  \or
    \@addamp
  \or
    \@acolampacol
  \or
    \@firstampfalse\@acol
  \fi
  \edef\@preamble{%
    \@preamble{%
      \ifcase\@chnum
        \hfil\ignorespaces\@sharp\unskip\hfil
```

```
      \or
        \hskip1sp\ignorespaces\@sharp\unskip\hfil
      \or
        \hfil\hskip1sp\ignorespaces\@sharp\unskip
      \fi}}}
\def\@classi{%
  \ifcase\@lastchclass
    \@acol\@arrayrule
  \or
    \@addtopreamble{\hskip \doublerulesep}\@arrayrule
  \or
  \or
  \or
    \@arrayrule
  \or
    \@acol\@arrayrule
  \or
    \@arrayrule
  \fi}
\def\@classii{%
  \ifcase\@lastchclass
  \or
    \@addtopreamble{\hskip .5\arrayrulewidth}%
  \fi}
\def\@classiii{\ifcase \@lastchclass \@acolampacol \or
   \@addamp\@acol \or
   \or \or \@addamp \or
   \@acolampacol \or \@ampacol \fi}
\def\@tabclassiv{\@addtopreamble\@nextchar}
\def\@arrayclassiv{\@addtopreamble{$\@nextchar$}}
\def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
\@sharp\@endpbox}}
\def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}
\newcount\@chclass
\newcount\@lastchclass
\newcount\@chnum
\newdimen\arraycolsep
\newdimen\tabcolsep
\newdimen\arrayrulewidth
\newdimen\doublerulesep
\def\arraystretch{1}    % Default value.
\newbox\@arstrutbox
\def\@arstrut{%
  \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}
\def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
   \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}
```

```
\def\@testpach#1{\@chclass \ifnum \@lastchclass=\tw@ 4 \else
    \ifnum \@lastchclass=3 5 \else
     \z@ \if #1c\@chnum \z@ \else
                              \if #1l\@chnum \@ne \else
                              \if #1r\@chnum \tw@ \else
          \@chclass \if #1|\@ne \else
                    \if #1@\tw@ \else
                    \if #1p3 \else \z@ \@preamerr 0\fi
  \fi  \fi  \fi  \fi  \fi  \fi
\fi}
\def\hline{%
  \noalign{\ifnum0=`}\fi\hrule \@height \arrayrulewidth \futurelet
   \reserved@a\@xhline}
\def\@xhline{\ifx\reserved@a\hline
               \vskip\doublerulesep
               \vskip-\arrayrulewidth
             \fi
      \ifnum0=`{\fi}}
\def\vline{\vrule \@width \arrayrulewidth}
\def\cline#1{\@cline#1\@nil}
\def\@cline#1-#2\@nil{%
  \omit
  \@multicnt#1%
  \advance\@multispan\m@ne
  \ifnum\@multicnt=\@ne\@firstofone{&\omit}\fi
  \@multicnt#2%
  \advance\@multicnt-#1%
  \advance\@multispan\@ne
  \leaders\hrule\@height\arrayrulewidth\hfill
  \cr
  \noalign{\vskip-\arrayrulewidth}}
\def\multispan{\omit\@multispan}
\def\@multispan#1{%
  \@multicnt#1\relax
  \loop\ifnum\@multicnt>\@ne \sp@n\repeat}
\def\sp@n{\span\omit\advance\@multicnt\m@ne}
```

`\@startpbox` `\@endpbox` Helper macros for 'p' columns.

`\@startpbox`{⟨*width*⟩} *text* `\egroup` is essentially `\parbox`{⟨*width*⟩}{⟨*text*⟩}

`\@endpbox` is essentially `\unskip \strut \par \egroup\hfil`

```
\def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
\def\@endpbox{\@finalstrut\@arstrutbox\par\egroup\hfil}
```

(*End definition for* `\@startpbox` *and* `\@endpbox`.)

\@@startpbox
\@@endpbox

```
\let\@@startpbox=\@startpbox
\let\@@endpbox=\@endpbox
```

(*End definition for* `\@@startpbox` *and* `\@@endpbox`.)

```
%</2ekernel>
```

phd package and give a background as to how fonts are used in LaTeX.

28.1 Introduction

When TEX was developed, the notion of graphic output was very limited, although Knuth presented a method using boxes to draw primitive commands at any point on the page. This of course is achieved using zero width or height **\hbox** or **\vbox** commands. LaTeX uses a similar approach with the picture environment. The **picture** environment comes straight out of the box and can be used to draw simple figures. For more sophisticated graphics **TikZ** is a better choice. It can be used in package documentation and simple tasks. The learning curve for using it is minimal.

Using the picture environment is much easier to code for drawing shapes or rulers around sectioning commands. This type of heading is very popular in many modern books. Figure **??**

Example 1.3: A corrosion study was made in order to determine whether coating an aluminum metal with a corrosion retardation substance reduced the amount of corrosion. The coating is a protectant that is advertised to minimize fatigue damage in this type of material. Also of interest is the influence of humidity on the amount of corrosion. A corrosion measurement can be expressed in thousands of cycles to failure. Two levels of coating, no coating and chemical corrosion coating, were used. In addition, the two relative humidity levels are 20% relative humidity and 80% relative humidity.

Figure 28.1: A section with some fancy lines around the text. From *Probabilities and Statistics for Engineers and Scientists*, by Walpole *et.al*, 2011.

Of course this is also achievable without the picture environment, simpy using TeX commands or with tikZ. From graphics point of view, the environment is also useful for short mathematical diagrams.

28.2 The Basic Commands

\begin{picture}
⟨*environment content*⟩
\end{picture}

The **picture** environment is created using one of two commands.

```
\begin{picture}(x, y). . . \end{picture}
```

or

```
\begin{picture}(x, y)(x0,y0). . . \end{picture}
```

\unitlength{⟨*dim*⟩}

Most people prefer the first type which they combine, with a **setlength** command that sets the **\unitlength**.

The optional argument gives the coordinates of the point at the lower-left corner of the picture (thereby determining the origin). For example, if **\unitlength** has been set to 1mm, the command

Example 2.1

```
\setlength\unitlength{1mm}
\begin{picture}(40,40)(0,0)
  \put(10,30){\vector(0,-1){30}}
  \put(10,30){\vector(1,0){30}}
  \put(25,30.5){$a$}
\end{picture}
```

produces a picture of width 100 millimeters and height 200 millimeters, whose lower-left corner is the point (10,20) and whose upper-right corner is therefore the point (110,220). When you first draw a picture, you will omit the optional argument, leaving the origin at the lower-left corner. If you then want to modify your picture by shifting everything, you just add the appropriate optional argument.

28.3 Text and Formulae

Text and formulas can be written into a picture environment with the **??**→ P. ?? command in the usual way. The line thickness can be set by using **??**→ P. ??{⟨*dim*⟩}. The command **\thinlines** is half the thickness of the **\linethickness** dimension and **\thicklines** is the current line width. The **\linethickness** does not change width of slanted lines or circles as it is drawn using a font and would render badly.

Example 3.2 Text and Formulae

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
 \thicklines
 \put(1,0.5){\line(2,1){3}}
 \put(4,2){\line(-2,1){2}}
 \put(2,3){\line(-2,-5){1}}
 \put(0.7,0.3){$A$}
 \put(4.05,1.9){$B$}
```

```
\put(1.7,2.95){$C$}
\put(3.1,2.5){$a$}
\put(1.3,1.7){$b$}
\put(2.5,1.05){$c$}
\put(0.3,4){$F=
\sqrt{s(s-a)(s-b)(s-c)}$}
\put(3.5,0.4){$\displaystyle
s:=\frac{a+b+c}{2}$}
\end{picture}
```

28.4 multiput and linethickness

The **\multiput** is used to place multiple objects onto the picture. It has the general format shown below:

$$F = \sqrt{s(s-a)(s-b)(s-c)}$$

Figure 28.2: Figures can have captions, if you enclose in a figure environment

Figure 28.3: Pictures can be scaled using \scalebox.

`\multiput(x,y) (Dx, Dy) {⟨n⟩} {⟨object⟩}`

The command `\multiput` allows to repeat a `\put` a number of times.

28.5 Some examples

Any vertex-symmetric graph is regular, but edge-symmetric graphs need not be regular. For example,

```
$$\unitlength=10pt
\def\putdisk(#1,#2){\put(#1,#2){\disk{.4}}}
$\vcenter{
\hbox{\beginpicture(2,1.5)(0,0)
\putdisk(0,0)
\putdisk(2,0)
\putdisk(1,.5)
\putdisk(1,1.5)
\put(0,0){\line(2,1){1}}
\put(2,0){\line(-2,1){1}}
```

```
\put(1,.5){\line(0,1){1}}
\endpicture}}
\quad&\hbox{is edge-symmetric, not vertex-symmetric;}\cr
\noalign{\smallskip}
\vcenter{
\hbox{\beginpicture(2,2)(0,0)
\putdisk(1,0)
\putdisk(1,2)
\putdisk(0,.5)
\putdisk(0,1.5)
\putdisk(2,.5)
\putdisk(2,1.5)
\put(0,.5){\line(2,1){2}}
\put(2,.5){\line(-2,1){2}}
\put(0,.5){\line(2,3){1}}
\put(2,.5){\line(-2,3){1}}
\put(0,.5){\line(1,0){2}}
\put(0,1.5){\line(1,0){2}}
\put(1,0){\line(-2,3){1}}
\put(1,0){\line(2,3){1}}
\put(1,0){\line(0,1){2}}
\endpicture}}
\quad&\hbox{is vertex-symmetric, not edge-symmetric.}\qquad
 (\vcenter{\hbox{\beginpicture(1,2)(0,0)
\putdisk(.5,0)\putdisk(.5,2)\put(.5,0){\line(0,1){2}}\endpicture}}
\hbox{ is a maximal clique})\cr}$$
```

28.6 picture package

The **picture** package by Heiko Oberdiek redefines the default **picture** macros and adds code that detects whether such an argument is given as number or as length. In the latter case, the length is used directly without multiplying with `\unitlength`. The following example is from the documentation of the package.

Example 6.3 Picture package example

```
\setlength{\unitlength}{1pt}
\begin{picture}(\widthof{Hello World}, 10mm)
  \put(0, 0){\makebox(0,0)[lb]{Hello World}}%
  \put(0, \heightof{Hello World} + \fboxsep){%
  \line(1, 0){\widthof{Hello World}}%
}%
\put(\widthof{Hello World}, 10mm){%
  \line(0, -1){10mm}%
}%
\put(0,0){\line(966,259){8}}
\end{picture}
```

The package **calc** is used for calculations or etex. The picture package requires that the package **calc** is loaded before the **picture** package and is loaded correctly by **phd**.

The package also supports the packages **pspicture** and **pict2e**, but they must be loaded before package picture.

28.7 pict2e

The package pict2e by Hubert Gäßlein, Rolf Nieprasch and Joseph Tkadlec extends the existing LATEX picture environment, using the familiar technique (cf. the graphics and color packages) of driver files. In the user-level part of this documentation there is a fair number of examples of use, showing where things are improved by comparison with the standard LaTeX picture environment.

The package is loaded automatically by **phd**.

28.8 xpicture

The package **xpicture** introduces several new graphicalinstructions, and some enriched versions of standard instructions used inside the `picture` environment. The picture environemnt establishes a rectangular coordinate system, so that all graphic objects are placed in the picture using the canonical coordinates of the plane.

phd package and give a background as to how fonts are used in LaTeX.

Picture mode commands. In addition to the commands available in LaTeX2.09, This section adds the new **\qbezier** command for drawing curves.

\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**

\shortstack{⟨*POS*⟩}{⟨*TEXT*⟩} : Makes a **\vbox** containing TEXT stacked as a one-column array, positioned l, r or c as indicated by POS.

Example 0.1 shortstack

```
\shortstack{Predefined\\Process}
\hspace{1cm}
\shortstack[r]{Predefined\\Process}
\shortstack[c]{Predefined\\Process}
```

Predefined Process Predefined Process Predefined Process

The '2ekernel' code ensures that a **\usepackage{autopict}** is essentially ignored if a 'full' format is being used that has picture mode already in the format.

```
%<2ekernel>\expandafter\let\csname ⤸
      ⤹ver@autopict.sty\endcsname\fmtversion
```

\@wholewidth
\@halfwidth

```
%<*2ekernel>
\newdimen\@wholewidth
\newdimen\@halfwidth
```

(*End definition for* `\@wholewidth` *and* `\@halfwidth`.)

\unitlength

```
\newdimen\unitlength \unitlength =1pt
```

(*End definition for* `\unitlength`.)

\@picbox
\@picht

```
\newbox\@picbox
\newdimen\@picht
```

(*End definition for* `\@picbox` *and* `\@picht`.)

\begin{picture}
⟨*environment content*⟩
\end{picture}

This is the main environment for drawing picture macros.

```
\long\gdef\picture#1{\pictur@#1}
\gdef\pictur@(#1){%
  \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}
```

\picture@
\@picture

```
\gdef\@picture(#1,#2)(#3,#4){%
  \@picht#2\unitlength
  \setbox\@picbox\hb@xt@#1\unitlength\bgroup
    \hskip -#3\unitlength
    \lower #4\unitlength\hbox\bgroup
      \ignorespaces}
```

(*End definition for* `\picture@` *and* `\@picture`.)

\endpicture

```
\gdef\endpicture{%
  \egroup\hss\egroup
    \ht\@picbox\@picht\dp\@picbox\z@
    \mbox{\box\@picbox}}
```

(*End definition for* `\endpicture`.)

In the definitions of **\put** and **\multiput**, **\hskip** was replaced by **\kern** just in case arg **#3** = "plus". (Bug detected by Don Knuth. changed 20 Jul 87).

```
\long\gdef\put(#1,#2)#3{%
  \@killglue\raise#2\unitlength
  \hb@xt@\z@{\kern#1\unitlength #3\hss}%
  \ignorespaces}
```

\multiput

```
\gdef\multiput(#1,#2)#3{%
  \@xdim #1\unitlength
  \@ydim #2\unitlength
   \@multiput(}
```

(*End definition for* `\multiput`.)

\multiput

```
\long\gdef\@multiput(#1,#2)#3#4{%
  \@killglue\@multicnt #3\relax
  \@whilenum \@multicnt >\z@\do
    {\raise\@ydim\hb@xt@\z@{\kern\@xdim #4\hss}%
     \advance\@multicnt\m@ne
     \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
  \ignorespaces}
```

(*End definition for* `\multiput`.)

\@killglue

```
\gdef\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}
```

(*End definition for* `\@killglue`.)

\thinlines
\thicklines

```
\gdef\thinlines{\let\@linefnt\tenln \let\@circlefnt\tencirc
  \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
\gdef\thicklines{\let\@linefnt\tenlnw \let\@circlefnt\tencircw
  \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}
```

(*End definition for* `\thinlines` *and* `\thicklines`.)

\linethickness

```
\gdef\linethickness#1{\@wholewidth #1\relax \@halfwidth .5\@wholewidth}
```

(*End definition for* `\linethickness`.)

\ishortstack

```
\gdef\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}
```

(*End definition for* `\ishortstack`.)

\@ishortstack

```
\gdef\@shortstack[#1]{%
  \leavevmode
  \vbox\bgroup
    \baselineskip-\p@\lineskip 3\p@
    \let\mb@l\hss\let\mb@r\hss
    \expandafter\let\csname mb@#1\endcsname\relax
    \let\\\@stackcr
    \@ishortstack}
```

(*End definition for* `\@ishortstack`.)

\@ishortstack

```
\gdef\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\crcr}\egroup}
```

(*End definition for* `\@ishortstack`.)

\@stackcr
\@ixstackcr

```
\gdef\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
\gdef\@ixstackcr{\@ifnextchar[\@istackcr{\cr\ignorespaces}}
```

(*End definition for* `\@stackcr` *and* `\@ixstackcr`.)

\@istackcr

```
\gdef\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}
```

(*End definition for* `\@istackcr`.)

\if@negarg

```
\newif\if@negarg
```

(*End definition for* `\if@negarg`.)

\line

The reason why we cannot use TEX's **\line**

```
\gdef\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@linelen #3\unitlength
  \ifdim\@linelen<\z@\@badlinearg\else
    \ifnum\@xarg =\z@ \@vline
      \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
    \fi
  \fi}
```

\@sline

```
\gdef\@sline{%
  \ifnum\@xarg<\z@ \@negargtrue \@xarg -\@xarg \@yyarg -\@yarg
  \else \@negargfalse \@yyarg \@yarg \fi
\ifnum \@yyarg >\z@ \@tempcnta\@yyarg \else \@tempcnta -\@yyarg \fi
\ifnum\@tempcnta>6 \@badlinearg\@tempcnta\z@ \fi
\ifnum\@xarg>6 \@badlinearg\@xarg \@ne \fi
\setbox\@linechar\hbox{\@linefnt\@getlinechar(\@xarg,\@yyarg)}%
```

If we have something like **\line(5,5){30}** the **\@linechar** will not contain a char and later on we will end in an infinite loop. So we check the width of the box and put in something as an emergency fix if necessary.

```
\ifdim\wd\@linechar=\z@
   \setbox\@linechar\hbox{.}%
   \@badlinearg
\fi
\ifnum \@yarg >\z@ \let\@upordown\raise \@clnht\z@
   \else\let\@upordown\lower \@clnht \ht\@linechar\fi
\@clnwd \wd\@linechar
\if@negarg
  \hskip -\wd\@linechar \def\reserved@a{\hskip -2\wd\@linechar}%
\else
     \let\reserved@a\relax
\fi
\@whiledim \@clnwd <\@linelen \do
  {\@upordown\@clnht\copy\@linechar
   \reserved@a
   \advance\@clnht \ht\@linechar
   \advance\@clnwd \wd\@linechar}%
\advance\@clnht -\ht\@linechar
\advance\@clnwd -\wd\@linechar
\@tempdima\@linelen\advance\@tempdima -\@clnwd
\@tempdimb\@tempdima\advance\@tempdimb -\wd\@linechar
\if@negarg \hskip -\@tempdimb \else \hskip \@tempdimb \fi
\multiply\@tempdima \@m
\@tempcnta \@tempdima
\@tempdima \wd\@linechar \divide\@tempcnta \@tempdima
\@tempdima \ht\@linechar \multiply\@tempdima \@tempcnta
\divide\@tempdima \@m
\advance\@clnht \@tempdima
```

```
\ifdim \@linelen <\wd\@linechar
   \hskip \wd\@linechar
   \ifdim \@linelen = \z@
   \else
     \@picture@warn
   \fi
   \else\@upordown\@clnht\copy\@linechar\fi}
```

(*End definition for* `\@sline`.)

`\@hline`

```
\gdef\@hline{\ifnum \@xarg <\z@ \hskip -\@linelen \fi
\vrule \@height \@halfwidth \@depth \@halfwidth \@width ↱
      ↳\@linelen
\ifnum \@xarg <\z@ \hskip -\@linelen \fi}
```

(*End definition for* `\@hline`.)

`\getlinechar`

```
\gdef\@getlinechar(#1,#2){\@tempcnta#1\relax\multiply\@tempcnta ↱
      ↳8%
  \advance\@tempcnta -9\ifnum #2>\z@ \advance\@tempcnta ↱
        ↳#2\relax\else
  \advance\@tempcnta -#2\relax\advance\@tempcnta 64 \fi
  \char\@tempcnta}
```

(*End definition for* `\getlinechar`.)

`\vector`

```
\gdef\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
  \ifnum\@tempcnta<5\relax
  \@linelen #3\unitlength
  \ifdim\@linelen<\z@\@badlinearg\else
    \ifnum\@xarg =\z@ \@vvector
      \else \ifnum\@yarg =\z@ \@hvector \else \@svector\fi
    \fi
  \fi
  \else\@badlinearg\fi}
```

(*End definition for* `\vector`.)

`\@hvector`

```
\gdef\@hvector{\@hline\hb@xt@\z@{\@linefnt
 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
    \hss\@getrarrow(1,0)\fi}}
```

(*End definition for* `\@hvector`.)

`\@vvector`

```
\gdef\@vvector{\ifnum \@yarg <\z@ \@downvector \else ↱
      ↳\@upvector \fi}
```

(*End definition for* `\@vvector`.)

`\@svector`

```
\gdef\@svector{\@sline
  \@tempcnta\@yarg \ifnum\@tempcnta <\z@ \@tempcnta -\@tempcnta\fi
  \ifnum\@tempcnta <5%
    \hskip -\wd\@linechar
    \@upordown\@clnht \hbox{\@linefnt  \if@negarg
    \@getlarrow(\@xarg,\@yyarg)\else \@getrarrow(\@xarg,\@yyarg)\fi}%
  \else\@badlinearg\fi}
```

(*End definition for* `\@svector`.)

`\@getlarrow`

```
\gdef\@getlarrow(#1,#2){\ifnum #2=\z@ \@tempcnta 27 % '33
  \else
  \@tempcnta #1\relax\multiply\@tempcnta \sixt@@n
  \advance\@tempcnta -9 \@tempcntb #2\relax\multiply\@tempcntb \tw@
  \ifnum \@tempcntb >\z@ \advance\@tempcnta \@tempcntb
  \else\advance\@tempcnta -\@tempcntb\advance\@tempcnta 64
  \fi\fi\char\@tempcnta}
```

(*End definition for* `\@getlarrow`.)

`\@getrarrow`

```
\gdef\@getrarrow(#1,#2){\@tempcntb #2\relax
\ifnum\@tempcntb <\z@ \@tempcntb -\@tempcntb\relax\fi
\ifcase \@tempcntb\relax \@tempcnta 45  % '55
\or
\ifnum #1<\thr@@ \@tempcnta #1\relax\multiply\@tempcnta
24\advance\@tempcnta -6 \else \ifnum #1=\thr@@ \@tempcnta 49
\else\@tempcnta 58 \fi\fi\or
\ifnum #1<\thr@@ \@tempcnta=#1\relax\multiply\@tempcnta
24\advance\@tempcnta -\thr@@ \else \@tempcnta 51 \fi\or
\@tempcnta #1\relax\multiply\@tempcnta
\sixt@@n \advance\@tempcnta -\tw@ \else
\@tempcnta #1\relax\multiply\@tempcnta
\sixt@@n \advance\@tempcnta 7 \fi\ifnum #2<\z@ \advance\@tempcnta 64 \fi
\char\@tempcnta}
```

(*End definition for* `\@getrarrow`.)

`\@vline`

```
\gdef\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}
```

(*End definition for* `\@vline`.)

\@upline

```
\gdef\@upline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \@linelen \@depth \z@\hss}}
```

(*End definition for* `\@upline`.)

\@downline

```
\gdef\@downline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \z@ \@depth \@linelen \hss}}
```

(*End definition for* `\@downline`.)

\@upvector

```
\gdef\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}% '66
  \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}
```

(*End definition for* `\@upvector`.)

\@downvector

```
\gdef\@downvector{\@downline\lower \@linelen
      \hb@xt@\z@{\@linefnt\char 63  % '77
      \hss}}
```

(*End definition for* `\@downvector`.)

\dashbox

```
\gdef\dashbox#1(#2,#3){\leavevmode\hb@xt@\z@{\baselineskip \z@skip
\lineskip \z@skip
\@dashdim #2\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
\ifodd\@dashcnt\@dashdim \z@
\advance\@dashcnt \@ne \divide\@dashcnt \tw@
\else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
\advance\@dashcnt \m@ne
\setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
\@width \@dashdim}\put(0,0){\copy\@dashbox}%
\put(0,#3){\copy\@dashbox}%
\put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
\put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
\multiply\@dashdim \thr@@
\fi
\setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
\@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
\put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
\do{\copy\@dashbox\advance\@tempcnta \@ne }}\@tempcnta\z@
```

```
\put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
\do{\copy\@dashbox\advance\@tempcnta \@ne }}%
\@dashdim #3\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
\ifodd\@dashcnt \@dashdim \z@
\advance\@dashcnt \@ne \divide\@dashcnt \tw@
\else
\divide\@dashdim \tw@ \divide\@dashcnt \tw@
\advance\@dashcnt \m@ne
\setbox\@dashbox\hbox{\hskip -\@halfwidth
\vrule \@width \@wholewidth
\@height \@dashdim}\put(0,0){\copy\@dashbox}%
\put(#2,0){\copy\@dashbox}%
\put(0,#3){\lower\@dashdim\copy\@dashbox}%
\put(#2,#3){\lower\@dashdim\copy\@dashbox}%
\multiply\@dashdim \thr@@
\fi
\setbox\@dashbox\hbox{\vrule \@width \@wholewidth
\@height #1\unitlength}\@tempcnta\z@
\put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
\do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}\@tempcnta\z@
\put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
\do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}\@makepicbox(#2,#3)}
```

(*End definition for* `\dashbox`.)

29.1 Circles and ovals

`\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.

`\if@ovt`
`\if@ovb`
`\if@ovl`
`\if@ovr`

If producing the Top Bottom Left or Right of an oval.

```
\newif\if@ovt
\newif\if@ovb
\newif\if@ovl
\newif\if@ovr
```

(*End definition for* `\if@ovt` *and others.*)

\@ovxx
\@ovyy
\@ovdx
\@ovdy
\@ovro
\@ovri

```
\newdimen\@ovxx
\newdimen\@ovyy
\newdimen\@ovdx
\newdimen\@ovdy
\newdimen\@ovro
\newdimen\@ovri
```

(*End 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

```
\gdef\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
  \@tempcnta\@tempdima
  \@tempdima 4\p@ \divide\@tempcnta\@tempdima
  \ifnum \@tempcnta >10\relax
      \@picture@warn
      \@tempcnta 10\relax
  \fi
  \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
```

Warn if requirements for oval or circle can't be met.[1]

```
    \else \@picture@warn \fi
  \multiply\@tempcnta 4\relax
  \setbox \@tempboxa \hbox{\@circlefnt
  \char \@tempcnta}\@tempdima \wd \@tempboxa}
```

(*End 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.

```
\def\@picture@warn{\@latex@warning{%
     \string\oval, \string\circle, or \string\line\space
     size unavailable}}
```

(*End definition for* `\@picture@warn`.)

\@put

```
\gdef\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}
```

(*End definition for* `\@put`.)

\oval

```
\gdef\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2)[]}}
```

(*End definition for* `\oval`.)

[1] v1.1g 1997/09/15 Warn if lines become invisible pr/2524

```
%</2ekernel>
%<latexrelease>\IncludeInRelease{2016/03/31}%
%<latexrelease>                 {\@ovhlinetrue}%
%<latexrelease>                 {Avoid almost zero length leaders}%
%<*2ekernel|latexrelease>
```

\if@ovvline
\if@ovhline

Tests whether horizontal or vertical lines are needed.

```
\newif\if@ovvline \@ovvlinetrue
\newif\if@ovhline \@ovhlinetrue
```

(*End definition for* `\if@ovvline` *and* `\if@ovhline`.)

\@oval

```
\gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
  \@ovvlinefalse \@ovhlinefalse
  \@tfor\reserved@a :=#3\do{\csname @ov\reserved@a false\endcsname}%
  \@ovxx #1\unitlength
  \@ovyy #2\unitlength
  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
  \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue \fi\fi
  \advance \@tempdimb -2\p@
  \@getcirc \@tempdimb
  \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
  \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
  \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
  \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
  \ifdim \@ovdy >\z@ \@ovvlinetrue \fi
  \@circlefnt \setbox\@tempboxa
  \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
  \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
  \if@ovt \@ovhorz \kern -\@ovxx \fi
  \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
  \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
  \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
  \endgroup}
```

(*End definition for* `\@oval`.)

\@ovvert

```
\gdef\@ovvert#1#2{\vbox to\@ovyy{%
    \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
      \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
    \else \kern \@ovri \kern \@ovdy \fi
    \if@ovvline \leaders\vrule \@width \@wholewidth \fi
```

```
    \vfil \nointerlineskip
    \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
      \hbox{\char \@tempcntb}%
    \else \kern \@ovdy \kern \@ovro \fi}}
```

(*End definition for* `\@ovvert`.)

\@ovhorz

```
\gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
    \if@ovr \else \kern \@ovdx \fi
    \if@ovhline \leaders \hrule \@height \@wholewidth \fi
    \hfil
    \if@ovl \else \kern \@ovdx \fi
    \kern \@ovri}}
```

(*End definition for* `\@ovhorz`.)

```
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\@ovhlinetrue}%
%<latexrelease>                 {Avoid almost zero length leaders}%
%<latexrelease>\let\if@ovvline\@undefined
%<latexrelease>\let\if@ovhline\@undefined
%<latexrelease>\gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
%<latexrelease>  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
%<latexrelease>  \@tfor\reserved@a :=#3\do
%<latexrelease>                {\csname @ov\reserved@a false\endcsname}%
%<latexrelease>  \@ovxx #1\unitlength
%<latexrelease>  \@ovyy #2\unitlength
%<latexrelease>  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
%<latexrelease>  \advance \@tempdimb -2\p@
%<latexrelease>  \@getcirc \@tempdimb
%<latexrelease>  \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
%<latexrelease>  \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
%<latexrelease>  \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
%<latexrelease>  \@circlefnt \setbox\@tempboxa
%<latexrelease>  \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
%<latexrelease>  \if@ovl
%<latexrelease>   \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
%<latexrelease>  \fi
%<latexrelease>  \if@ovt \@ovhorz \kern -\@ovxx \fi
%<latexrelease>  \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
%<latexrelease>  \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
```

```
%<latexrelease>  \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
%<latexrelease>  \endgroup}
%<latexrelease>\gdef\@ovvert#1#2{\vbox to\@ovyy{%
%<latexrelease>    \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
%<latexrelease>      \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
%<latexrelease>    \else \kern \@ovri \kern \@ovdy \fi
%<latexrelease>    \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
%<latexrelease>    \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
%<latexrelease>      \hbox{\char \@tempcntb}%
%<latexrelease>    \else \kern \@ovdy \kern \@ovro \fi}}
%<latexrelease>\gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
%<latexrelease>    \if@ovr \else \kern \@ovdx \fi
%<latexrelease>    \leaders \hrule \@height \@wholewidth \hfil
%<latexrelease>    \if@ovl \else \kern \@ovdx \fi
%<latexrelease>    \kern \@ovri}}
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

`\circle`

```
\gdef\circle{\@inmatherr\circle\@ifstar\@dot\@circle}
```

(*End definition for* `\circle`.)

`\@circle`

```
\gdef\@circle#1{%
  \begingroup \boxmaxdepth \maxdimen \@tempdimb #1\unitlength
   \ifdim \@tempdimb >15.5\p@ \@getcirc\@tempdimb
      \@ovro\ht\@tempboxa
     \setbox\@tempboxa\hbox{\@circlefnt
      \advance\@tempcnta\tw@ \char \@tempcnta
      \advance\@tempcnta\m@ne \char \@tempcnta \kern -2\@tempdima
      \advance\@tempcnta\tw@
      \raise \@tempdima \hbox{\char\@tempcnta}\raise \@tempdima
        \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
      \@put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
   \else  \@circ\@tempdimb{96}\fi\endgroup}
```

(*End definition for* `\@circle`.)

`\@dot` Internal form of `\circle*`.

```
\gdef\@dot#1{\@tempdimb #1\unitlength \@circ\@tempdimb{112}}
```

(*End definition for* `\@dot`.)

`\@circ`

```
\gdef\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
   \@tempcnta\@tempdima \@tempdima \p@
```

```
    \divide\@tempcnta\@tempdima
    \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
    \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
    \advance\@tempcnta #2\relax
    \@circlefnt \char\@tempcnta}
```

(*End definition for* `\@circ`.)

```
\@xarg
\@yarg \newcount\@xarg
\@yyarg \newcount\@yarg
        \newcount\@yyarg
```

(*End definition for* `\@xarg`, `\@yarg`, *and* `\@yyarg`.)

```
\@multicnt
        \newcount\@multicnt
```

(*End definition for* `\@multicnt`.)

```
\@xdim
\yxdim \newdimen\@xdim
       \newdimen\@ydim
```

(*End definition for* `\@xdim` *and* `\yxdim`.)

```
\@linechar
        \newbox\@linechar
```

(*End definition for* `\@linechar`.)

```
\@linelen
        \newdimen\@linelen
```

(*End definition for* `\@linelen`.)

```
\@clnwd
\@clnht \newdimen\@clnwd
        \newdimen\@clnht
```

(*End definition for* `\@clnwd` *and* `\@clnht`.)

```
\@dashdim
\@dashbox \newdimen\@dashdim
\@dashcnt \newbox\@dashbox
          \newcount\@dashcnt
```

(*End definition for* `\@dashdim`, `\@dashbox`, *and* `\@dashcnt`.)

Initialization: "`\thinlines`" [2]

```
\let\@linefnt\tenln
\let\@circlefnt\tencirc
\@wholewidth\fontdimen8\tenln
\@halfwidth .5\@wholewidth
```

[2] v1.1f 1995/10/27 Move initialisation to kernel from autoload file

29.2 Curves

The new **\qbezier** command, based on the old **\bezier** defined in **bezier.sty**.[3]

\qbeziermax The maximum number of points to plot.

```
\gdef\qbeziermax{500}
```

(*End definition for* `\qbeziermax`.)

\qbezier

```
\newcommand\qbezier[2][0]{\bezier{#1}#2}
```

(*End definition for* `\qbezier`.)

\bezier

```
\gdef\bezier#1)#2(#3)#4({\@bezier#1)(#3)(}
```

\@bezier

```
\gdef\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
  \ifnum #1=\z@
      \@ovxx #4\unitlength
        \advance\@ovxx -#2\unitlength
        \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
      \@ovdx #6\unitlength
        \advance\@ovdx -#4\unitlength
        \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
        \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
      \@ovyy #5\unitlength
        \advance\@ovyy -#3\unitlength
        \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
      \@ovdy #7\unitlength
        \advance\@ovdy -#5\unitlength
        \ifdim \@ovdy<\z@  \@ovdy -\@ovdy \fi
        \ifdim \@ovyy<\@ovdy \@ovyy  \@ovdy \fi
      \@multicnt
         \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
      \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
      \ifnum \qbeziermax<\@multicnt ↩
           ↪\@multicnt\qbeziermax\relax \fi
  \else \@multicnt#1\relax \fi
  \@tempcnta\@multicnt \advance\@tempcnta\@ne
  \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
      \multiply\@ovdx \tw@
  \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
      \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
  \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
       \multiply\@ovdy \tw@
  \@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
      \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt
```

[3] v0.1c 1994/04/28 bezier curves added

Fix pr/3566[4]

```
 \setbox\@tempboxa\hbox{%
           \hskip -\@halfwidth
           \vrule \@height\@halfwidth
                  \@depth \@halfwidth
                  \@width \@wholewidth}%
  \put(#2,#3){%
    \count@\z@
    \@whilenum{\count@<\@tempcnta}\do
       {\@xdim\count@\@ovxx
          \advance\@xdim\@ovdx
          \divide\@xdim\@multicnt
          \multiply\@xdim\count@
        \@ydim\count@\@ovyy
          \advance\@ydim\@ovdy
          \divide\@ydim\@multicnt
          \multiply\@ydim\count@
        \raise \@ydim
          \hb@xt@\z@{\kern\@xdim
                     \unhcopy\@tempboxa\hss}%
        \advance\count@\@ne}}}
%</2ekernel>
```

(*End definition for* `\bezier` *and* `\@bezier`.)

[4] v1.1k 2003/08/27 added missing displacement pr/3566

phd package and give a background as to how fonts are used in LaTeX.

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*⟩.

Example 0.1 newtheorem

```
\newtheorem{theorem}{Theorem}[section]

\newtheorem{lemma}[theorem]{Lemma}

%\newtheorem{proposition}[theorem]{Proposition}
%\newtheorem{corollary}[theorem]{Corollary}
%\newenvironment{proof}[1][Proof]{\begin{trivlist}
%\item[\hskip \labelsep {\bfseries #1}]}{\end{trivlist}}
\newenvironment{definition}[1][Definition]{\begin{trivlist}
\item[\hskip \labelsep {\bfseries #1}]}{\end{trivlist}}

%\newenvironment{example}[1][Example]{\begin{trivlist}
%\item[\hskip \labelsep {\bfseries #1}]}{\end{trivlist}}

%\newenvironment{remark}[1][Remark]{\begin{trivlist}
%\item[\hskip \labelsep {\bfseries #1}]}{\end{trivlist}}
%
%\newcommand{\qed}{\nobreak \ifvmode \relax \else
%      \ifdim\lastskip<1.5em \hskip-\lastskip
%      \hskip1.5em plus0em minus0.5em \fi \nobreak
%      \vrule height0.75em width0.5em depth0.25em\fi}
\begin{theorem}
\begin{definition}
Let $H$ be a subgroup of a group~$G$.  A \emph{left coset}
of $H$ in $G$ is a subset of $G$ that is of the form $xH$,
where $x \in G$ and $xH = \{ xh : h \in H \}$.
Similarly a \emph{right coset} of $H$ in $G$ is a subset
of $G$ that is of the form $Hx$, where
$Hx = \{ hx : h \in H \}$
\end{definition}

Note that a subgroup~$H$ of a group $G$ is itself a
```

```
left coset of $H$ in $G$.

\begin{lemma}
\label{LeftCosetsDisjoint}
Let $H$ be a subgroup of a group $G$, and let $x$ and $y$ be
elements of $G$.  Suppose that $xH \cap yH$ is non-empty.
Then $xH = yH$.
\end{lemma}

\begin{proof}
Let $z$ be some element of $xH \cap yH$.  Then $z = xa$
for some $a \in H$, and $z = yb$ for some $b \in H$.
If $h$ is any element of $H$ then $ah \in H$ and
$a^{-1}h \in H$, since $H$ is a subgroup of $G$.
But $zh = x(ah)$ and $xh = z(a^{-1}h)$ for all $h \in H$.
Therefore $zH \subset xH$ and $xH \subset zH$, and thus
$xH = zH$.  Similarly $yH = zH$, and thus $xH = yH$,
as required.\qed
\end{proof}

\begin{lemma}
\label{SizeOfLeftCoset}
Let $H$ be a finite subgroup of a group $G$.  Then each left
coset of $H$ in $G$ has the same number of elements as $H$.
\end{lemma}

\begin{proof}
Let $H = \{ h_1, h_2,\ldots, h_m\}$, where
$h_1, h_2,\ldots, h_m$ are distinct, and let $x$ be an
element of $G$.  Then the left coset $xH$ consists of
the elements $x h_j$ for $j = 1,2,\ldots,m$.
Suppose that $j$ and $k$ are integers between
$1$ and $m$ for which $x h_j = x h_k$.  Then
$h_j = x^{-1} (x h_j) = x^{-1} (x h_k) = h_k$,
and thus $j = k$, since $h_1, h_2,\ldots, h_m$
are distinct.  It follows that the elements
$x h_1, x h_2,\ldots, x h_m$ are distinct.
We conclude that the subgroup~$H$ and the left
coset $xH$ both have $m$ elements,
as required.
\end{proof}

\begin{theorem}
\emph{(Lagrange's Theorem)}
\label{Lagrange}
Let $G$ be a finite group, and let $H$ be a subgroup
of $G$.  Then the order of $H$ divides the order of $G$.
```

```
\end{theorem}

\begin{proof}
Each element~$x$ of $G$ belongs to at least one left coset
of $H$ in $G$ (namely the coset $xH$), and no element
can belong to two distinct left cosets of $H$ in $G$
(see Lemma~\ref{LeftCosetsDisjoint}).  Therefore every
element of $G$ belongs to exactly one left coset of $H$.
Moreover each left coset of $H$ contains $|H|$ elements
(Lemma~\ref{SizeOfLeftCoset}).  Therefore $|G| = n |H|$,
where $n$ is the number of left cosets of $H$ in $G$.
The result follows.
\end{proof}
\end{theorem}
```

Theorem 30.0.1. ***Definition*** *Let H be a subgroup of a group G. A* left coset *of H in G is a subset of G that is of the form xH, where $x \in G$ and $xH = \{xh : h \in H\}$. Similarly a* right coset *of H in G is a subset of G that is of the form Hx, where $Hx = \{hx : h \in H\}$*
Note that a subgroup H of a group G is itself a left coset of H in G.

Lemma 30.0.2. *Let H be a subgroup of a group G, and let x and y be elements of G. Suppose that $xH \cap yH$ is non-empty. Then $xH = yH$.*

Proof. Let z be some element of $xH \cap yH$. Then $z = xa$ for some $a \in H$, and $z = yb$ for some $b \in H$. If h is any element of H then $ah \in H$ and $a^{-1}h \in H$, since H is a subgroup of G. But $zh = x(ah)$ and $xh = z(a^{-1}h)$ for all $h \in H$. Therefore $zH \subset xH$ and $xH \subset zH$, and thus $xH = zH$. Similarly $yH = zH$, and thus $xH = yH$, as required. □ □

Lemma 30.0.3. *Let H be a finite subgroup of a group G. Then each left coset of H in G has the same number of elements as H.*

Proof. Let $H = \{h_1, h_2, \ldots, h_m\}$, where $h_1, h_2, \ldots, h_m$ are distinct, and let x be an element of G. Then the left coset xH consists of the elements xh_j for $j = 1, 2, \ldots, m$. Suppose that j and k are integers between 1 and m for which $xh_j = xh_k$. Then $h_j = x^{-1}(xh_j) = x^{-1}(xh_k) = h_k$, and thus $j = k$, since $h_1, h_2, \ldots, h_m$ are distinct. It follows that the elements $xh_1, xh_2, \ldots, xh_m$ are distinct. We conclude that the subgroup H and the left coset xH both have m elements, as required. □

Theorem 30.0.4. (Lagrange's Theorem) *Let G be a finite group, and let H be a subgroup of G. Then the order of H divides the order of G.*

Proof. Each element x of G belongs to at least one left coset of H in G (namely the coset xH), and no element can belong to two distinct left cosets of H in G (see Lemma **??**). Therefore every element of G belongs to exactly one left coset of H. Moreover each left coset of H contains `H` elements (Lemma **??**). Therefore `G` $= n$`H`, where n is the number of left cosets of H in G. The result follows. □

Example from source

\newtheorem

\newtheorem ought really be allowed only in the preamble Which would be good document style, and allow some main memory to be saved by declaring these commands to be \@onlypreamble. Unfortunately the LaTeX book indicates that \newtheorem may be used anywhere in the document...

```
%<*2ekernel>
\def\newtheorem#1{%
  \@ifnextchar[{\@othm{#1}}{\@nthm{#1}}}
```

(*End definition for* \newtheorem.)

\@nthm

```
\def\@nthm#1#2{%
  \@ifnextchar[{\@xnthm{#1}{#2}}{\@ynthm{#1}{#2}}}
```

(*End definition for* \@nthm.)

\@xnthm

```
\def\@xnthm#1#2[#3]{%
  \expandafter\@ifdefinable\csname #1\endcsname
    {\@definecounter{#1}\@newctr{#1}[#3]%
     \expandafter\xdef\csname the#1\endcsname{%
       \expandafter\noexpand\csname the#3\endcsname ↩
            ↪\@thmcountersep
          \@thmcounter{#1}}%
     \global\@namedef{#1}{\@thm{#1}{#2}}%
     \global\@namedef{end#1}{\@endtheorem}}}
```

(*End definition for* \@xnthm.)

\@ynthm

```
\def\@ynthm#1#2{%
  \expandafter\@ifdefinable\csname #1\endcsname
    {\@definecounter{#1}%
     \expandafter\xdef\csname ↩
          ↪the#1\endcsname{\@thmcounter{#1}}%
     \global\@namedef{#1}{\@thm{#1}{#2}}%
     \global\@namedef{end#1}{\@endtheorem}}}
```

(*End definition for* \@ynthm.)

\@othm

```
\def\@othm#1[#2]#3{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}%
    {\expandafter\@ifdefinable\csname #1\endcsname
    {\global\@namedef{the#1}{\@nameuse{the#2}}%
  \global\@namedef{#1}{\@thm{#2}{#3}}%
  \global\@namedef{end#1}{\@endtheorem}}}}
```

(*End definition for* \@othm.)

\@thm

```
\def\@thm#1#2{%
  \refstepcounter{#1}%
  \@ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
```

(*End definition for* `\@thm`.)

\@xthm
\@ythm

```
\def\@xthm#1#2{%
  \@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}
\def\@ythm#1#2[#3]{%
  \@opargbegintheorem{#2}{\csname ⤸
       ⤹the#1\endcsname}{#3}\ignorespaces}
```

(*End definition for* `\@xthm` *and* `\@ythm`.)

Default values

\@thmcounter
\@thmcountersep

```
\def\@thmcounter#1{\noexpand\arabic{#1}}
\def\@thmcountersep{.}
```

(*End definition for* `\@thmcounter` *and* `\@thmcountersep`.)

\@begintheorem
\@opargbegintheorem
\@endtheorem

Providing theorem defaults.

```
\def\@begintheorem#1#2{\trivlist
   \item[\hskip \labelsep{\bfseries #1\ #2}]\itshape}

\def\@opargbegintheorem#1#2#3{\trivlist
      \item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]\itshape}
\def\@endtheorem{\endtrivlist}
%</2ekernel>
```

(*End definition for* `\@begintheorem`, `\@opargbegintheorem`, *and* `\@endtheorem`.)

30.1 AmS Improvements

The package **amsthm** provides numerous facilities to authors to enable styling of theorem-like environments to be customized.

phd package and give a background as to how fonts are used in LaTeX.

31.1 Introduction

This file defines the declarations such as **\author** which are used by **\maketitle**. **\maketitle** itself is defined by each class, not in the LaTeX kernel.

The second part of the file defines the generic commands used for defining sectioning commands such as **\chapter**. Again the actual document level commands are defined in the class files, in terms of these commands.

The final part provides a framework for adding material to a table of contents. The basic author commands provided here are shown below. Most of the balance commands are aimed at the class or package writer.

Table 31.1: User commands defined in file ltsec.

\title	defines the title and author
\author	author name
\thanks	adds thanks as a footnote
\and	
\date	typesets a date

```
%<*2ekernel>
\message{title,}
```

31.2 The Title

\title **\author** The user defines the title and author by the declarations **\title**{⟨*name*⟩}, **\author**{⟨*name*⟩}

\date Similarly the date is declared with **\date**{⟨*date*⟩}.

\thanks **\and** Inside these, the **\thanks**{⟨*footnote text*⟩} command may be used to make acknowledgements, notice of address, etc. in a footnote. If there are multiple authors, they have to be separated with the **\and** command.

\maketitle And finally, the **\maketitle** command produces the actual title, using the information previously saved with the other commands.

\title **\@title** [1] **\title** for use in **\maketitle**. If not given **\maketitle** will produce an error message.

```
\def\title#1{\gdef\@title{#1}}
\def\@title{\@latex@error{No \noexpand\title given}\@ehc}
```

(*End definition for* `\title` *and* `\@title`.)

\author **\@author** [2]

\author for use in **\maketitle**. If not given **\maketitle** will produce a warning message.

[1] LaTeX2e 1993/12/11 Added default
[2] LaTeX2e 1993/12/11 Added default

```
\def\author#1{\gdef\@author{#1}}
\def\@author{\@latex@warning@no@line{No \noexpand\author ↪
      ↪given}}
```

(*End definition for* `\author` *and* `\@author`.)

`\date`
`\@date`

`\date` for use in `\maketitle`. If not given `\maketitle` will produce `\today` as the default.

```
\def\date#1{\gdef\@date{#1}}
\gdef\@date{\today}
```

(*End definition for* `\date` *and* `\@date`.)

[3]

`\thanks`

```
\def\thanks#1{\footnotemark
    \protected@xdef\@thanks{\@thanks
        \protect\footnotetext[\the\c@footnote]{#1}}%
}
```

(*End definition for* `\thanks`.)

`\@thanks`

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

(*End definition for* `\@thanks`.)

`\and`

```
\def\and{%                    % \begin{tabular}
  \end{tabular}%
  \hskip 1em \@plus.17fil%
  \begin{tabular}[t]{c}}%     % \end{tabular}
```

(*End definition for* `\and`.)

```
\message{sectioning,}
```

31.3 Sectioning

`\@secpenalty`

```
\newcount\@secpenalty
\@secpenalty = -300
```

(*End definition for* `\@secpenalty`.)

`\if@noskipsec`
`\@noskipsectrue`

Way back in 1991 (08/26) FMi & RmS set the `\@noskipsec` switch to true for the preamble and to false in `\document`. This was done to trap lists and related text in the preamble but it does not catch everything.

[3] 1.0h 1994/11/04 (ASAJ) Added `\protected@xdef` to `\thanks`.

```
\newif\if@noskipsec \@noskipsectrue
```

(*End 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 ...
            \paragraph{...}
          \endgroup}
```

Pseudocode for the `\@startsection` command

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
  \@tempskipa #4\relax
  \@afterindenttrue
  \ifdim \@tempskipa <\z@
    \@tempskipa -\@tempskipa \@afterindentfalse
  \fi
  \if@nobreak
    \everypar{}%
  \else
```

```
    \addpenalty\@secpenalty\addvspace\@tempskipa
  \fi
  \@ifstar
    {\@ssect{#3}{#4}{#5}{#6}}%
    {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}
```

(*End definition for* `\@startsection`.)

`\@sect`

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
    \let\@svsec\@empty
  \else
    \refstepcounter{#1}%
```

Since `\@seccntformat` might end with an improper `\hskip` which is scanning forward for `plus` or `minus` we end the definition of `\@svsec` with `\relax` as a precaution.

```
    \protected@edef\@svsec{\@seccntformat{#1}\relax}%
  \fi
  \@tempskipa #5\relax
  \ifdim \@tempskipa>\z@
    \begingroup
      #6{%
        \@hangfrom{\hskip #3\relax\@svsec}%
          \interlinepenalty \@M #8\@@par}%
    \endgroup
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{%
      \ifnum #2>\c@secnumdepth \else
        \protect\numberline{\csname the#1\endcsname}%
      \fi
      #7}%
  \else
    \def\@svsechd{%
      #6{\hskip #3\relax
      \@svsec #8}%
      \csname #1mark\endcsname{#7}%
      \addcontentsline{toc}{#1}{%
        \ifnum #2>\c@secnumdepth \else
          \protect\numberline{\csname the#1\endcsname}%
        \fi
        #7}}%
  \fi
  \@xsect{#5}}
```

(*End definition for* `\@sect`.)

`\@xsect`

```
\def\@xsect#1{%
  \@tempskipa #1\relax
  \ifdim \@tempskipa>\z@
```

```
    \par \nobreak
    \vskip \@tempskipa
    \@afterheading
  \else
    \@nobreakfalse
    \global\@noskipsectrue
    \everypar{%
      \if@noskipsec
        \global\@noskipsecfalse
       {\setbox\z@\lastbox}%
        \clubpenalty\@M
        \begingroup \@svsechd \endgroup
        \unskip
        \@tempskipa #1\relax
        \hskip -\@tempskipa
      \else
        \clubpenalty \@clubpenalty
        \everypar{}%
      \fi}%
  \fi
  \ignorespaces}
```

(*End definition for* `\@xsect`.)

`\@seccntformat` This command formats the section number including the space following it.

```
\def\@seccntformat#1{\csname the#1\endcsname\quad}
```

(*End definition for* `\@seccntformat`.)

`\@ssect`

```
\def\@ssect#1#2#3#4#5{%
  \@tempskipa #3\relax
  \ifdim \@tempskipa>\z@
    \begingroup
      #4{%
        \@hangfrom{\hskip #1}%
          \interlinepenalty \@M #5\@@par}%
    \endgroup
  \else
    \def\@svsechd{#4{\hskip #1\relax #5}}%
  \fi
  \@xsect{#3}}
```

(*End definition for* `\@ssect`.)

`\if@afterindent`
`\@afterindenttrue`

```
\newif\if@afterindent \@afterindenttrue
```

(*End definition for* `\if@afterindent` *and* `\@afterindenttrue`.)

`\@afterheading` This hook is used in setting up custom-built headings in classes.dtx.

```
\def\@afterheading{%
  \@nobreaktrue
  \everypar{%
    \if@nobreak
      \@nobreakfalse
      \clubpenalty \@M
      \if@afterindent \else
        {\setbox\z@\lastbox}%
      \fi
    \else
      \clubpenalty \@clubpenalty
      \everypar{}%
    \fi}}
```

(*End 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.

```
\def\@hangfrom#1{\setbox\@tempboxa\hbox{{#1}}%
      \hangindent \wd\@tempboxa\noindent\box\@tempboxa}
```

(*End definition for* `\@hangfrom`.)

`\c@secnumdepth`
`\c@tocdepth`

```
\newcount\c@secnumdepth
\newcount\c@tocdepth
```

(*End 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*{`...`}`

```
\def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

(*End definition for* `\secdef`.)

31.3.1 Initializations

`\sectionmark`
`\subsectionmark`
`\subsubsectionmark`
`\paragraphmark`
`\subparagraphmark`

```
\let\sectionmark\@gobble
\let\subsectionmark\@gobble
\let\subsubsectionmark\@gobble
\let\paragraphmark\@gobble
\let\subparagraphmark\@gobble
```

(*End definition for* `\sectionmark` *and others.*)

```
\message{contents,}
```

31.4 Table of Contents etc.

31.4.1 Convention

`\tf@`⟨*foo*⟩ = file number for output for table foo. The file is opened only if `@filesw` = `true`.

31.4.2 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.

```
\def\@starttoc#1{%
  \begingroup
    \makeatletter
    \@input{\jobname.#1}%
    \if@filesw
      \expandafter\newwrite\csname tf@#1\endcsname
      \immediate\openout \csname tf@#1\endcsname ↩
            ↪\jobname.#1\relax
    \fi
    \@nobreakfalse
  \endgroup}
```

(*End definition for* `\@starttoc`.)

`\addcontentsline` The `\addcontentsline`{⟨*table*⟩}{⟨*type*⟩}{⟨*entry*⟩} command allows the user to add his/her own entry to a table of contents, etc. The command adds the entry `\contentsline`{⟨*type*⟩}{⟨*entry*⟩}{⟨*page*⟩} to the `.`⟨*table*⟩ file.

This macro is implemented as an application of `\addtocontents`. Note that `\thepage` is not expandable during `\protected@write` therefore one gets the page number at the time of the `\shipout`.

```
%</2ekernel>
```

```
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2018/12/01}%
%<latexrelease>                 {\addcontentsline}{Mask line endings}%
\def\addcontentsline#1#2#3{%
  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}%
                     \protected@file@percent}}
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\addcontentsline}{Mask line endings}%
%<latexrelease>\def\addcontentsline#1#2#3{%
%<latexrelease>  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

(*End definition for* `\addcontentsline`.)

`\addtocontents` The `\addtocontents{`⟨*table*⟩`}{`⟨*text*⟩`}` command adds ⟨*text*⟩ to the `.`⟨*table*⟩ file, with no page number.

```
\long\def\addtocontents#1#2{%
  \protected@write\@auxout
      {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
      {\string\@writefile{#1}{#2}}}
```

(*End definition for* `\addtocontents`.)

`\contentsline` The `\contentsline{`⟨*type*⟩`}{`⟨*entry*⟩`}{`⟨*page*⟩`}` macro produces a ⟨*type*⟩ entry in a table of contents, etc. It will appear in the `.toc` or other file. For example, The entry for subsection 1.4.3 in the table of contents for example, might be produced by:

```
\contentsline{subsection}
    {\makebox{30pt}[r]{1.4.3} Gnats and Gnus}{22}
```

The `\protect` command causes command sequences to be written without expanding them.

```
\def\contentsline#1{\csname l@#1\endcsname}
```

(*End definition for* `\contentsline`.)

`\@dottedtocline{`⟨*level*⟩`}{`⟨*indent*⟩`}{`⟨*numwidth*⟩`}{`⟨*title*⟩`}{`⟨*page*⟩`}`: Macro to produce a table of contents line with the following parameters:

level If ⟨*level*⟩ > `\c@tocdepth`, then no line produced.

indent Total indentation from the left margin.

numwidth Width of box for number if the ⟨*title*⟩ has a `\numberline` command. As of 25 Jan 1988, this is also the amount of extra indentation added to second and later lines of a multiple line entry.

title Contents of entry.

page Page number.

Uses the following parameters, which must be set by the document style. They should be defined with `\def`'s.

`pnumwidth` Width of box in which page number is set.

`tocrmarg` Right margin indentation for all but last line of multiple-line entries.

`dotsep` Separation between dots, in mu units. Should be `\def`'d to a number like 2 or 1.7

`\@dottedtocline`

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2018/12/01}%
%<latexrelease>                 {\@dottedtocline}{Prevent protrusion}%
\def\@dottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth \else
    \vskip \z@ \@plus.2\p@
    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
     \parindent #2\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima #3\relax
```

Changes[4]

```
     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {#4}\nobreak
     \leaders\hbox{$\m@th
```

If a document uses fonts other than computer modern, the use of a dot from math can be very disturbing despite the fact that this might be the only place in a document that then uses computer modern. Therefore we surround the dot with an `\hbox` to escape to the surrounding text font.

```
        \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
        mu$}\hfill
     \nobreak
     \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5%
```

We finish off by preventing any protrusion if that is enabled. If protrusion happens the number may shift to the right and as a result you may end up with an additional dot in the toc line in some situations.

[4] v1.0z 1996/12/20 Added `\nobreak` for latex/2343

```
                          \kern-\p@\kern\p@}%
      \par}%
   \fi}
```

(*End definition for* `\@dottedtocline`.)

`\noprotrusion` This command, if placed directly to the right (or left) of a word, will prevent prevent protrusion of that word into the margin. It is used in the toc entry lines as they shouldn't protrude. It is implemented as to kerns that cancel each other but being there hide the word so that protrusion is not added. Note that a zero kern or an empty box would not work as the protrusion mechanism will skip over those.

```
\DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}
```

(*End definition for* `\noprotrusion`.)

```
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\@dottedtocline}{Prevent protrusion}%
%<latexrelease>\def\@dottedtocline#1#2#3#4#5{%
%<latexrelease>  \ifnum #1>\c@tocdepth \else
%<latexrelease>    \vskip \z@ \@plus.2\p@
%<latexrelease>    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
%<latexrelease>     \parindent #2\relax\@afterindenttrue
%<latexrelease>     \interlinepenalty\@M
%<latexrelease>     \leavevmode
%<latexrelease>     \@tempdima #3\relax
%<latexrelease>     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
%<latexrelease>     {#4}\nobreak
%<latexrelease>     \leaders\hbox{$\m@th
%<latexrelease>        \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
%<latexrelease>        mu$}\hfill
%<latexrelease>     \nobreak
%<latexrelease>     \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
%<latexrelease>     \par}%
%<latexrelease>  \fi}
%<latexrelease>
%<latexrelease>\let\noprotrusion\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

Note: `\nobreak`'s added 7 Jan 86 to prevent bad line break that left the page number dangling by itself at left edge of a new line.

Changed 25 Jan 88 to use `\leftskip` instead of `\hangindent` so leaders of multiple-line contents entries would line up properly.

`\numberline` `\numberline{`⟨*number*⟩`}`: For use in a `\contentsline` command. It puts ⟨*number*⟩ flushleft in a box of width `\@tempdima` (Before 25 Jan 88 change, it also added `\@tempdima` to the hanging indentation.)

```
\def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
%</2ekernel>
```

(*End definition for* `\numberline`.)

phd package and give a background as to how fonts are used in LaTeX.

32.1 Float types

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. See //tex.stackexchange.com/questions/39017/how-to-influence-the-positio 39020#39020

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:

! : bang — indicates that some restrictions should be ignored
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. The order of specifying the placement specifiers is irrelevant to the float algorithm.

However, if a letter is not present then the corresponding area will not be tried at all.

32.1.1 Floating Environments

```
\message{floats,}
```

Where floats may appear on a page, and how many may appear there are specified by the following float placement parameters. The numbers are named like counters so the user can set them with the ordinary counter-setting commands.

`\c@topnumber`	Number of floats allowed at the top of a column.
`\topfraction`	Fraction of column that can be devoted to floats. `\c@dbltopnumber`, `\dbltopfraction` Same as above, but for double-column floats.
`\c@bottomnumber`, `\bottomfraction`	Same as above for bottom of page.
`\c@totalnumber`	Number of floats allowed in a single column, including in-text floats.
`\textfraction`	Minimum fraction of column that must contain text.

`\floatpagefraction`	Minimum fraction of page that must be taken up by float page.
`\dblfloatpagefraction`	Same as above, for double-column floats.

The document class 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`NUMTEXT	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.

Example 1.1 Type parameters

```
\makeatletter
\fps@figure \\
\ftype@figure
\makeatother
```

tbp
1

The command is designed to be called by the class in a new environment definition, for example in AmS classes figure is defined as:

The `\@float` will take care of the basic structure of the float and to set penalties so as to be able to pass this information to the `output` routine.

```
\@float{TYPE}[PLACEMENT] ==
 BEGIN
   if hmode then \@bsphack
                          \@floatpenalty := -10002
             else         \@floatpenalty := -10003
   fi
   \@captype ==L TYPE
   \@dblflset
   \@fps     ==L PLACEMENT
```

```
    \@onelevel@sanitize \@fps
    add default PLACEMENT if at most ! in PLACEMENT ==
         \@fpsadddefault
    if inner
      then LaTeX Error: 'Not in outer paragraph mode.'
           \@floatpenalty := 0
      else if \@freelist nonempty
             then \@currbox  :=L head of \@freelist
                  \@freelist :=G tail of \@freelist
                  \count\@currbox :=G 32*\ftype@TYPE +
                                          bits determined by
                                               PLACEMENT
             else \@floatpenalty := 0
                  LaTeX Error: 'Too many unprocessed floats'
           fi
    fi
    \@currbox :=G   \color@vbox
                      \normalcolor
                        \vbox{
                         %% 15 Dec 87 --
                         %% removed \boxmaxdepth :=L 0pt
                         %% that made box 0 depth because
                              it screwed
                         %% things up. Instead, added
                              \vskip0pt at end
                              \hsize = \columnwidth
                              \@parboxrestore
                              \@floatboxreset
  END

 \caption ==
   BEGIN
    \refstepcounter{\@captype}
    \@dblarg{\@caption{\@captype}}
   END

In following definition, \par moved from after
      \addcontentsline to
before \addcontentsline because the \write could cause
an extra blank line to be added to the paragraph above the
caption.  (Change made 12 Jun 87)

 \@caption{TYPE}[STEXT]{TEXT} ==
  BEGIN
    \par
    \addcontentsline{\ext@TYPE}{TYPE}{\numberline{\theTYPE}{STEXT}}
    \begingroup
      \@parboxrestore
      \@normalsize
      \@makecaption{\fnum@TYPE}{TEXT}
      \par
    \endgroup
  END

```

```

  \@dblfloat{TYPE}[PLACEMENT] : Macro to begin a float environment for
    a double-column float of type TYPE with PLACEMENT as the placement
    specifier.  The default value of PLACEMENT is 'tp'
    The environment is ended by \end@dblfloat.
    E.g., \figure* == \@dblfloat{figure},
          \endfigure* == \end@dblfloat.

  \@dblfloat{TYPE}[PLACEMENT] ==
    Identical to \@float{TYPE}[PLACEMENT] except \hsize and \linewidth
    are set to \textwidth.
```

\@floatpenalty The float penalty is saved in a counter to enable ease of use.

(*End definition for* `\@floatpenalty`.)

```
\newcount\@floatpenalty
```

\caption[⟨*short caption text*⟩]{⟨*long caption text*⟩}

This is set to be an error message outside a float since no captype is defined there; this may need to be changed by some classes. Note if the caption is outside a float it triggers an error.

```
\def\caption{%
  \ifx\@captype\@undefined
    \@latex@error{\noexpand\caption outside float}\@ehd
    \expandafter\@gobble
  \else
    \refstepcounter\@captype
    \expandafter\@firstofone
  \fi
  {\@dblarg{\@caption\@captype}}%
}
```

The paragraph setting parameters are normalised at this point, however `\@parboxrestore` resets `\everypar` which is not correct in this context so `\@setminipage` is called if needed.

The float mechanism, like minipage, sets the flag `@minipage` true before executing the user-supplied text. Many LaTeX constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates TeX's 'top of page' behaviour. The flag must be set false at the start of the first paragraph. This is achieved by a redefinition of `\everypar`, but the call to `\@parboxrestore` removes that redefinition, so it is re-inserted if needed. If the flag is already false then the `\caption` was not the first entry in the float, and so some other paragraph has already activated the special `\everypar`. In this case no further action is needed.

```
    \@parboxrestore
    \if@minipage
      \@setminipage
    \fi
```

```
    \normalsize
    \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
  \endgroup}
```

\@float{⟨*float specifier*⟩}

Just a reminder that this is used to define new floating environments, such as *figure* and *table*. The **fps@#1** has been defined in the standard classes and defines the default float specifier. In **book.cls** is defined as **\def\fps@figure{tbp}**.

Example 1.2 The definition of float

```
\makeatletter
\meaning\fps@figure
\makeatother
```

macro:->tbp

\@dblflset

```
\def\@float#1{%
  \@ifnextchar[%
    {\@xfloat{#1}}%
    {\edef\reserved@a{\noexpand\@xfloat{#1}[\csname fps@#1\endcsname]}%
     \reserved@a}}
```

\@dblfloat

(*End definition for* \@dblfloat.)

```
\def\@dblfloat{%
  \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
  \reserved@a}
```

\fps@dbl Note that all double floats have default fps 'tp'.

(*End definition for* \fps@dbl.)

\@setfps This sets the fps, dealing with error conditions by adding the default.

(*End 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.

The **\@nodocument** is defined in the ltxerror and is the error produced if paragraphs are typeset in the preamble. So here LaTeX is checking if a float was specified in a preamble. Why here, is because we cannot rely on user defined environments to carry out these tests.

What does the ! do? This goes to default, so it is interesting to find out why the bang was specified in the first place. It just starts overrides all the constraints.

(*End definition for* `\@xfloat`.)

```
\def\@xfloat #1[#2]{%
  \@nodocument
  \def \@captype {#1}%
   \def \@fps {#2}%
   \@onelevel@sanitize \@fps
   \def \reserved@b {!}%
   \ifx \reserved@b \@fps
     \@fpsadddefault
   \else
     \ifx \@fps \@empty
       \@fpsadddefault
     \fi
   \fi
%    \end{macrocode}
% As discussed earlier, floats will set different penalties if it is in horizontal mode
% from that of other modes.
%    \begin{macrocode}
   \ifhmode
     \@bsphack
     \@floatpenalty -\@Mii
   \else
     \@floatpenalty-\@Miii
   \fi
% If inner obviously we cannot float out of a box, issue an error
  \ifinner
     \@parmoderr\@floatpenalty\z@
  \else
%
    \@next\@currbox\@freelist
      {%
       % holds the type of float start at 16. Note the same count is used in
       % the output routine to set the bits.
       \@tempcnta \sixt@@n

       % iterate through the specification and set the bitmap
       \expandafter \@tfor \expandafter \reserved@a
         \expandafter :\expandafter =\@fps
         \do
          {%
           % This is the first position reading from the right.
           \if \reserved@a h%
```

```
              \ifodd \@tempcnta
              \else
                % 17 is here float
                \advance \@tempcnta \@ne
              \fi
            \fi
            \if \reserved@a t%
              % position 2
              \@setfpsbit \tw@

            \fi
            \if \reserved@a b%

              \@setfpsbit 4%
            \fi
            \if \reserved@a p%
              \@setfpsbit 8%
            \fi
            \if \reserved@a !%
              \ifnum \@tempcnta>15
                \advance\@tempcnta -\sixt@@n\relax
              \fi
            \fi
            }%
        \@tempcntb \csname ftype@\@captype \endcsname

        % multiply by 32 by the ftype\@captype
        \multiply \@tempcntb \@xxxii
        \advance \@tempcnta \@tempcntb

        % define globally count |\@currbox| as |\@tempcnta|
        \global \count\@currbox \@tempcnta
        }%
    \@fltovf  %This is for too many floats error for marginpars
  \fi
```

The remainder sets up the box in which the float is typeset, and the typesetting environment to be used. It is essential to have the extra box to avoid the unwanted space that would otherwise often be put at the top of the float.

It ends with a hook; not sure how useful this is but it is needed at present to deal with double-column floats.

```
  \global \setbox\@currbox
    \color@vbox
      \normalcolor
      \vbox \bgroup
        \hsize\columnwidth
        \@parboxrestore
        \@floatboxreset
}
```

`\@floatboxreset` The rational for allowing these normally global flags to be set locally here, via `\@parboxrestore`, was stated originally by Donald Arseneau and extended by Chris Rowley. It is because these flags are only set globally to true by sec-

tion commands, and these should never appear within marginals or floats or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in **\set@nobreak**; otherwise this command will be redundant. Added local settings of flags: dangerous!!

(*End definition for* `\@floatboxreset`.)

```
\def \@floatboxreset {%
        \reset@font
        \normalsize
        \@setminipage
}
```

`\@setnobreak`

```
\def \@setnobreak{%
  \if@nobreak
    \let\outer@nobreak\@nobreaktrue
    \@nobreakfalse
  \fi
}
```

(*End definition for* `\@setnobreak`.)

`\@setminipage`

```
\def \@setminipage{%
  \@minipagetrue
  \everypar{\@minipagefalse\everypar{}}%
}
```

(*End definition for* `\@setminipage`.)

\end@float

```
\def\end@float{%
  \@endfloatbox
  \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed **\textheight**, otherwise float will never get typeset (91/03/15 FMi).

```
    \@largefloatcheck
    \@cons\@currlist\@currbox
    \ifnum\@floatpenalty <-\@Mii
      \penalty -\@Miv
```

Saving and restoring **\prevdepth** added 26 May 87 to prevent extra vertical space when used in vertical mode.

```
      \@tempdima\prevdepth
      \vbox{}%
      \prevdepth\@tempdima
      \penalty\@floatpenalty
```

```
    \else
      \vadjust{\penalty -\@Miv ↩
            ↪\vbox{}\penalty\@floatpenalty}\@Esphack
    \fi
  \fi
}
```

\end@dblfloat

```
\def\end@dblfloat{%
\if@twocolumn
  \@endfloatbox
  \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed **\textheight**, otherwise float will never get typeset (91/03/15 FMi).

```
    \@largefloatcheck
    \@cons\@dbldeferlist\@currbox
  \fi
```

RmS 92/03/18 changed **\@esphack** to **\@Esphack**.

```
    \ifnum \@floatpenalty =-\@Mii \@Esphack\fi
\else
  \end@float
\fi
}
```

(*End definition for* `\end@dblfloat`.)

\@endfloatbox

This macro is not intended to be a hook; it is designed to help maintain the integrity of this code, which is used twice and, as can be seen, is subject to frequent changes.

```
\def \@endfloatbox{%
      \par\vskip\z@skip      %% \par\vskip\z@ added 15 Dec 87
      \@minipagefalse
      \outer@nobreak
    \egroup                  %% end of vbox
  \color@endbox
}
```

\outer@nobreak

```
\let\outer@nobreak\@empty
```

(*End definition for* `\outer@nobreak`.)

\@largefloatcheck

(*End definition for* `\@largefloatcheck`.)

This calculates by how much a float is oversize for the page and prints this in a warning message.

```
\def \@largefloatcheck{%
  \ifdim \ht\@currbox>\textheight
    \@tempdima -\textheight
    \advance \@tempdima \ht\@currbox
    \@latex@warning {Float too large for page by ↊
         ↊\the\@tempdima}%
    \ht\@currbox \textheight
  \fi
}
```

\@dbflt
\@xdblfloat

```
\def\@dbflt#1{\@ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
\def\@xdblfloat#1[#2]{%
  \@xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}
```

(*End definition for* `\@dbflt` *and* `\@xdblfloat`.)

```
\newcount\c@topnumber
\newcount\c@dbltopnumber
\newcount\c@bottomnumber
\newcount\c@totalnumber
```

An analysis of **\@floatplacement**:

This should be called whenever **\@colht** has been set.

```
\def\@floatplacement{\global\@topnum\c@topnumber
    % Textpage bit, global:
   \global\@toproom \topfraction\@colht
   \global\@botnum  \c@bottomnumber
   \global\@botroom \bottomfraction\@colht
   \global\@colnum  \c@totalnumber
    % Floatpage bit, local:
   \@fpmin   \floatpagefraction\@colht}
```

\@dblfloatplacement

This should be called only within a group. Now changed to provide extra checks in **\@addtodblcol**, needed when processing a BANG float.

```
\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
  \global \@dbltopnum \c@dbltopnumber
  \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses **\@textmin** to locally store the amount of extra room in the column.

```
  \@textmin \@colht
  \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
  \@fpmin \dblfloatpagefraction\textheight
  \@fptop \@dblfptop
  \@fpsep \@dblfpsep
  \@fpbot \@dblfpbot
}
```

(*End definition for* `\@dblfloatplacement`.)

32.2 Marginal Notes

Marginal notes use the same mechanism as floats to communicate with the **??**[→P.??] 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,nd 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.

left
right

`\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. (I was unable to reproduce the error).

```
\marginpar [LTEXT]{RTEXT} ==
 BEGIN
   if hmode then \@bsphack
                 \@floatpenalty := -10002
            else \@floatpenalty := -10003
   fi
   if inner
     then LaTeX Error: 'Not in outer paragraph mode.'
          \@floatpenalty := 0
     else if \@freelist has two elements:
            then get \@marbox, \@currbox  from \@freelist
                 \count\@marbox :=G -1
            else \@floatpenalty := 0
                 LaTeX Error: 'Too many unprocessed floats'
                 \@currbox, \@marbox := \@tempboxa    %%use \def
          fi
   fi
   if optional argument
     then %% \@xmpar ==
          \@savemarbox\@marbox{LTEXT}
          \@savemarbox\@currbox{RTEXT}
     else %% \@ympar ==
```

```
                \@savemarbox\@marbox{RTEXT}
                \box\@currbox :=G \box\@marbox
        fi
        \@xympar
       END

    \reversemarginpar == BEGIN \@mparbottom   :=G 0
                               @reversemargin :=G true
                         END

    \normalmarginpar  == BEGIN \@mparbottom   :=G 0
                               @reversemargin :=G false
                         END
```

\marginpar

```
\def\marginpar{%
  \ifhmode
    \@bsphack
    \@floatpenalty -\@Mii
  \else
    \@floatpenalty-\@Miii
  \fi
  \ifinner
    \@parmoderr
    \@floatpenalty\z@
  \else
    \@next\@currbox\@freelist{}{}%
    \@next\@marbox\@freelist{\global\count\@marbox\m@ne}%
       {\@floatpenalty\z@
        \@fltovf\def\@currbox{\@tempboxa}\def\@marbox{\@tempboxa}}%
  \fi
  \@ifnextchar [\@xmpar\@ympar}
```

\@xmpar

```
\long\def\@xmpar[#1]#2{%
  \@savemarbox\@marbox{#1}%
  \@savemarbox\@currbox{#2}%
  \@xympar}
```

(*End definition for* `\@xmpar`.)

This is the main macro for a marginpar command that does not have left or right text. Note it call `\@xympar`

\@ympar

```
\long\def\@ympar#1{%
  \@savemarbox\@marbox{#1}%
  \global\setbox\@currbox\copy\@marbox
  \@xympar}
```

(*End definition for* `\@ympar`.)

\@savemarbox Sets up the vboxes including a color@vbox to correctly handle colour.

```
\long\def \@savemarbox #1#2{%
  \global\setbox #1%
    \color@vbox
      \vtop{%
        \hsize\marginparwidth
        \@parboxrestore
        \@marginparreset
        #2%
        \@minipagefalse
        \outer@nobreak
        }%
    \color@endbox
}
```

(*End definition for* `\@savemarbox`.)

\@marginparreset The rational for allowing these normally global flags to be set locally here, via **\@parboxrestore** was stated originally by Donald Arsenau and extended by Chris Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within marginals or floats or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in **\set@nobreak**; otherwise this command will be redundant.

```
\def \@marginparreset {%
        \reset@font
        \normalsize
%        \let\if@nobreak\iffalse
%        \let\if@noskipsec\iffalse
%        \@setnobreak
        \@setminipage
}
```

(*End definition for* `\@marginparreset`.)

\@xympar

```
\def \@xympar{%
  \ifnum\@floatpenalty <\z@\@cons\@currlist\@marbox\fi
  \setbox\@tempboxa
    \color@vbox
      \vbox \bgroup
  \end@float
  \@ignorefalse
  \@esphack
}
```

(*End definition for* `\@xympar`.)

\reversemarginpar
\normalmarginpar

```
\def\reversemarginpar{\global\@mparbottom\z@ ⤸
      ⤹\@reversemargintrue}
```

```
\def\normalmarginpar{\global\@mparbottom\z@ ⤸
      ⤹\@reversemarginfalse}
```

(*End definition for* `\reversemarginpar` *and* `\normalmarginpar`.)

```
\message{footnotes,}
```

32.3 Footnotes

We start with a summary of all user commands.

`\footnote`{⟨*text*⟩}	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`[⟨*number*⟩]

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`[⟨*number*⟩]{⟨*text*⟩}

Command to produce the footnote but no mark. `\footnote` is equivalent to `\footnotemark \footnotetext`.

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`

```
\footnote{NOTE}  ==
 BEGIN
   \stepcounter{\@mpfn}
   begingroup
      \protect == \noexpand
      \@thefnmark :=G eval (\thempfn)
   endgroup
   \@footnotemark
   \@footnotetext{NOTE}
 END
```

`\footins` LaTeX uses the same insert for footnotes as PLAIN.

```
\newinsert\footins
```

```
\footnotetext[NUM] ==
begin
  begingroup
  counter \@mpfn := L NUM
  \protect == \noexpand
  \@thefnmark :=G eval(\thempfn)
  endgroup
  \@footnotetext
end
```

LaTeX leaves these initializations for the `\footins` insert.

```
\skip\footins=\bigskipamount
\count\footins=1000
\dimen\footins=8in
```

(*End definition for* `\footins`.)

`\footnoterule` LaTeX keeps PLAIN TeX's `\footnoterule` as the default. This is also redefined in the standard classes.

```
\def\footnoterule{\kern-3\p@
  \hrule \@width 2in \kern 2.6\p@} % the \hrule is .4pt high
```

(*End definition for* `\footnoterule`.)

`\thefootnote`

```
\@definecounter{footnote}
\def\thefootnote{\@arabic\c@footnote}
```

(*End 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.

```
\@definecounter{mpfootnote}
\def\thempfootnote{{\itshape\@alph\c@mpfootnote}}
```

(*End definition for* `\thempfootnote`.)

`\@makefnmark`

```
\def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
```

(*End definition for* `\@makefnmark`.)

`\textsuperscript` This command provides superscript characters in the current text font. It's implementation might change!!!

```
\DeclareRobustCommand*\textsuperscript[1]{%
  \@textsuperscript{\selectfont#1}}
```

(*End definition for* `\textsuperscript`.)

\@textsuperscript

This command should not be used directly, but may be used to define other commands **\textsuperscript**, **\@makefnmark**. **#1** should always start with a font selection command, to activate the font size switch.

```
\def\@textsuperscript#1{%
  {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}}
```

(*End definition for* `\@textsuperscript`.)

\footnotesep

See the geometry chapter for the value of **\footnotesep**. In this book it was set as 6.65pt.

```
\newdimen\footnotesep
```

(*End definition for* `\footnotesep`.)

\footnote

The LaTeX **footnote** command takes an optional parameter.

```
\def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
    \protected@xdef\@thefnmark{\thempfn}%
    \@footnotemark\@footnotetext}}
```

As packages tend to modify almost everything in LaTeX I was curious to check the definition of footnote.

Example 3.3 The meaning of footnote

```
The meaning of footnote command.\footnote[13]{This is a
    footnote.}

\texttt{\meaning\footnote}
```

The meaning of footnote command.[m]

macro:->\@ifnextchar [\@xfootnote {\stepcounter \@mpfn \protected@xdef \@thefnmark {\thempfn }\@footnotemark \@footnotetext }

[m] This is a footnote.

\@xfootnote

```
\def\@xfootnote[#1]{%
  \begingroup
    \csname c@\@mpfn\endcsname #1\relax
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotemark\@footnotetext}
```

(*End definition for* `\@xfootnote`.)

\@footnotetext

```
\long\def\@footnotetext#1{\insert\footins{%
    \reset@font\footnotesize
    \interlinepenalty\interfootnotelinepenalty
    \splittopskip\footnotesep
    \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
    \hsize\columnwidth \@parboxrestore
    \protected@edef\@currentlabel{%
       \csname p@footnote\endcsname\@thefnmark
    }%
    \color@begingroup
      \@makefntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
    \color@endgroup}}%
```

\footnotemark[⟨*number*⟩]

The footnote mark is defined here. See also the book class.

```
\def\footnotemark{%
   \@ifnextchar[\@xfootnotemark
     {\stepcounter{footnote}%
      \protected@xdef\@thefnmark{\thefootnote}%
      \@footnotemark}}
```

\@xfootnotemark

```
\def\@xfootnotemark[#1]{%
   \begingroup
      \c@footnote #1\relax
      \unrestored@protected@xdef\@thefnmark{\thefootnote}%
   \endgroup
   \@footnotemark}
```

\@footnotemark

Add **\nobreak** to allow hyphenation. latex/1605

```
\def\@footnotemark{%
  \leavevmode
  \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
  \@makefnmark
  \ifhmode\spacefactor\@x@sf\fi
  \relax}
```

(*End definition for* `\@footnotemark`.)

\footnotetext

```
\def\footnotetext{%
     \@ifnextchar [\@xfootnotenext
       {\protected@xdef\@thefnmark{\thempfn}%
    \@footnotetext}}
```

(*End definition for* `\footnotetext`.)

\@xfootnotenext

```
\def\@xfootnotenext[#1]{%
  \begingroup
     \csname c@\@mpfn\endcsname #1\relax
     \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotetext}
```

(*End definition for* `\@xfootnotenext`.)

\thempfn
\@mpfn

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
```

(*End definition for* `\thempfn` *and* `\@mpfn`.)

phd package and give a background as to how fonts are used in LaTeX.

Sherborn's 'Index Animalium'. Credit Natural History Museum, London[a]

PREPARING INDEXES

And in such indexes, although small pricks
To their subsequent volumes, there is seen
The baby figure of the giant mass
Of things to come at large.
–Shakespeare
from
Troilus and Cressida

SELF-PORTRAIT Handwritten cards compiled by Sherborn for his publication *Index Animalium*

THE first English Language index, appeared in Christopher Marlowe's *Hero and Leander* in 1593. At that period, as often as not, by an "index to a book" was meant what we should now call a table of contents. Among the first indexes—in the modern sense—to a book in the English language was one in Plutarch's Parallel Lives, in Sir Thomas North's 1595 translation[b].

A section entitled "An Alphabetical Table of the most material contents of the whole book" may be found in Henry Scobell's Acts and Ordinances of Parliament of 1658. This section comes after "An index of the general titles comprised in the ensuing Table". Both of these indexes predate the index to Alexander Cruden's Concordance (1737) see Farrows[c], which is erroneously held to be the earliest index found in an English book.

[a] This monumental publication became the basis for all zoological nomenclature work having gathered together all the relevant data in one place, just as an online database does today.

[b] Borko, Harold & Bernier, Charles L. (1978). *Indexing Concepts and Methods*, ISBN 0-12-118660-1.

[c] **farrow96**.

33.1 Preparing an index

In order to produce an index, we need to load the package **makeidx**, mark the indices we need using `\index` and compile the doument using `\makeindex`.[1] At the place where we want the index to be printed,we use `\printindex`.

In LaTeX, we include a word in the index by using the command `\index`⟨*arg*⟩, so if the word Kroll should be included in the index, we should use the command `\index{Kroll}`.

If the word is to be printed in bold, we use

```
\documentclass{article}
\usepackage{makeidx}
\makeindex
\begin{document}
   To prepare an index,
   just include the
   package\index{package}
\printindex
\end{document}
```

When the input file is proceesed with one of the LaTeX engines the `\index` command writes an appropriate index with the current page number, to a special file. The file has the same name as the LaTeX input file, but with the extension (`.idx`). This file can them be processed with the `makeindex` program by typing in the command line:

```
makeindex filename
```

Note that you do not have to type the *filename* extension. The program will look for a filename.idx and use that. The `makeindex` program generates a sorted index with the same base fille name, but this time with the extension `.ind`. if now the LaTeX input file is processed again, this sorted index gets included into the document at the point where LaTeX finds `\printindex`.

There is a special character `@` which is used to denote that what appears on its left side must be typeset as it appears on its right side. The first occurrence of perl will also be used by the sorting algorithm. is is very useful since what is used for sorting and what will be printed may be different! For example,we may want to have the name `Donald Knuth` under the letter K., we should write

```
\index{Knuth@{Donald Knuth}}
```

Another thing we may want to change is the way that the page number is typeset. If we want, for example, to have the page number in bold, we would write `\index{perl|textbf}`. Notice that we wrote "textbf" without the backslash. Of course, the above can be combined with the command:

```
\index{perl@\textbf{perl}|textit}
```

will print the word perl in the index (the entry will be typeset in boldface type) sorted as "perl",' and its page number will be italic. A common application of this is through the command . If we want to send the reader to another index entry, say, to send the reader from the `$\omega$` to the `$\Omega$` command, we can write

```
\index{omega@$\omega$|see{$\Omega$}}
```

Here, we ask for the entry to be sorted according to the word omega and, in its place, the program must use

```
$\omega$|see{$\Omega$}
```

If a word is used repeatedly in a

[1] You will also need to at least run the file once using `MakeIndex` on `MikTex`. Check your distribution if you getting problems.

range of pages and we want to have this range in the index, we do not write the relative `index` command all of the time. Instead, we write `\index\{convex\|(\}` at the place where we have the first occurrence and `\index\{convex\|)\}` at the place where we have the last occurrence. This will produce a page range in the index for the word "convex".

33.1.1 Subindices

Subindices can be produced using an exclamation mark. If we want the word 'Zeus' to appear in the category of Greek which is in the category of Gods, we will write:

```
\index{Gods!Greek!Zeus}
```

The actual symbol used will depend on the `.ist` file used when the file is compiled. For example in the `ltxdoc` and **phddoc** classes this is redefined as `>` and the above should be written as:

```
\index{Gods>Greek>Zeus}
```

The sympbol to be used is called `index level` and is defined in the `.ist` file that is used in conjuction with the `MakeIndex` program to produce the index.

33.2 Multi-page Indexing

To perform multi-page indexing, add a `( and )` to the end of the `\index` command, as in .

```
\index{Quantum Mechanics!History|(}
```

> In 1901, Max Planck released his theory of radiation dependant on quantized energy. While this explained the ultraviolet catastrophe in the spectrum of blackbody radiation, this had far larger consequences as the beginnings of quantum mechanics....

```
\index{Quantum Mechanics>History|)}
```

33.3 Summary of commands

Example	Index Entry	Comment
\index{hello}	hello, 1	Plain entry
\index{hello!Peter}	Peter, 3	Subentry under 'hello'
\index{Sam@\textsl{Sam}}	Sam, 2	Formatted entry
\index{Lin@\textbf{Lin}}	**Lin**, 7	Same as above
\index{Jennytextbf}	Jenny, 3	Formatted page number
\index{Joe textit}	Joe, 5	Same as above
\index{ecole@école}	école, 4	Handling of accents
\index{Peter see{hello}}	Peter, see hello	Cross-references
\index{Jen see also{Jenny}}	Jen, see also Jenny	Same as above

33.4 Indexing Class Documentation

Indexing LaTeX 2e classes or package documentation produced with the **ltxdoc** is normally achieved using the **doc**[2] and **docstrip**[3] program, which are sometime difficult to use, if you need to deviate from their standard methods. The important thing here to remember is that you need to use different characters `= > *`.

normal	doc
@	=
!	>

```
\index{Indexing=\textbf}
\index{Indexing>general}
\index{Indexing>doc}
```

This manual, was build using a large **`ltxdoc`** class and these problems appeared while I was developing it. As normal with such problems, they were very time consuming to debug. There are still issues in some parts and one day, I am hoping to come back and correct them. One needs at this point to query the need to use the **`doc`** and **`docstrip`** method of documenting macros and if it shouldn't have a pre-processor written in a higher language to ease development.

Example 4.1 Checking usage and main

```
\meaning\main \\
\meaning\usage
```

macro:#1->{\color {red}\uline {#1}}
macro:#1->\textit {#1}

[2] Mittelbach, Frank (May 2018). *The doc and shortvrb package*.
[3] Mittelbach, Frank et al. (May 2018). *The DocStrip Program*. Version v2.5g.

phd package and give a background as to how fonts are used in LaTeX.

The makeindex program was developed by Pehong Chen and Michael Harrison[1] in the eighties and is still used by LaTeX. It is a remarkable, flexible program, found on most distributions. It is also used in GNU EMACS.

The input format consists of a list of ⟨*specifier, attribute*⟩ tuples. These are the essential tokens and delimiters needed in scanning the input index file. There are numerous such specifiers (you can think of them as the names of functions or variables) and the attributes as their value.

Table 34.1: Index input style parameters

S/No	Specifier	Attribute	Default	Meaning
4	`keyword`	string	`"\\indexentry"`	index command
5	`arg_open`	char	`'{'`	argument opening delimiter
6	`arg_close`	char	`'}'`	argument closing delimiter
7	`range_open`	char	`'('`	page range opening delimiter
8	`range_close`	char	`')'`	page range closing delimiter
9	`level`	char	`'!'`	index level delimiter
10	`actual`	char	`'@'`	actual key designator
11	`encap`	char	`|`	page number encapsulator
12	`quote`	char	`'"'`	quote symbol
13	`escape`	char	`'\\'`	symbol which escapes quote
14	`page_compositor`	string	"-"	composite page delimiter

page range The two range delimiters `range_open` and `range_close` are used with the `encap` operator. When `range_open` immediately follows `encap` (i.e., `\index{...|(...}`), it tells the index processor that an explicit range is starting. Conversely `range_close` signals the closing of a range. In our design, three or more successive page numbers are abbreviated as a range implicitly. This *implicit* range formation can be turned off if an indexed term represents logically distinct concepts in different pages. When the implicit range is disabled, *explicit* page ranges can be enforced by using the two range delimiters `range_open` and `range_close`. Therefore, it is possible to index an entire section or a large piece of text related to a certain concept without having to insert an index command in every single page.

quote The `quote` operator is used to escape symbols. Thus `\index{foo"@goo}` means a sort key of `foo@goo` rather than a sort key of `foo"` and an actual key of `goo`. As an exception, `quote`, when preceded by `escape` (i.e. `\index{...\"...}`), does not escape its succeeding letter. This special case is included because `\"` is the umlaut command in TeX. Requiring `quote` itself to be quoted in this case (i.e. `\""`) is feasible but somewhat awkward; `quote` and `escape` must be distinct.

[1] Chen, Pehong (1986). *GNU Emacs TeX -mode*. Tech. rep. 87/316. University of California. URL: http://digitalassets.lib.berkeley.edu/techreports/ucb/text/CSD-87-347.pdf.

composite page delimiter A page number can be a composite of one or more fields separated by a certain delimiter bound to a *page_compositor* e.g. II-13 for page 13 of Chapter II. This attribute allows the lexical analyzer of the makeindex program to seprate these fields, making the sorting of page numbers easier.

34.1 Output format

Table **??** summarizes the output style parameters. Again, it is a list of <*specifier*, *attribute*> pairs. In the default column, '`\n`' and '`\t`' denote a new line and a tab, respectively. These parameters can be further divided into the following groups:

Context . Together, `preamble` and `postamble` define the context in which the index is to be formatted.

Starting Page . The starting page number can either be supplied by the user or retrieved automatically from the document transcript. In either case, this number can be enclosed with `setpage_prefix` and `setpage_suffix` to yield a page number initializing command.

New Group/Letter . The string bound to `group_skip` denotes the extra vertical space needed when a group is started. For a group beginning with a different letter, the parameters `lethead_prefix` and `lethead_suffix` (both with a default nil string) denote the group heading. The flag `lethead_flag` has a default value of 0, which means other than `group_skip` nothing else will be inserted before the group. On the other hand, if this flag is positive, the strings bound to `lethead_prefix` and `lethead_suffix` will be inserted with an instance of the new letter in uppercase in between. Similarly, a lowercase letter will be inserted if the flag is negative.

Entry Separators . This group includes everything with the `item_` prefix. First, `item_`i denotes the command and indentation to be inserted when a key is started from a level greater than or equal to i. Second, `item_`ij has a similar meaning, but with $i = j - 1$. Finally, the two `item_x`j's are included to handle the situation where the parent level has no page numbers. Some styles require cases like these to be different from those with page numbers.

Table **??** depicts a system that supports three levels of subindexing. In general, suppose n is the number of index levels supported, there will be n `item_`i's $(0 \le i \le n-1)$, $(n-1)$ `item_`ij's $(1 \le j \le n-1, i = j-1)$, and $(n-1)$ `item_x`j's $(1 \le j \le n-1)$.

Page Delimiters Each level has a key/page delimiter that defines what is to be inserted between a key and its first page number. The inter-page delimiter is specified by `delim_n`, while the range designator is given by `delim_r`.

Page Encapsulator The attributes of `encap_prefix`, `encap_infix`, and `encap_suffix` form what is to be placed into the output when an en-

capsulator is specified for a certain entry. Suppose `foo` is the specified encapsulator and `N` is the page number, the output sequence is

`encap_prefix foo encap_infix N encap_suffix`

Page Precedence Five different types of numerals are supported by most systems for page numbering. These are lowercase roman (`r`), numeric or arabic (`n`), lowercase alphabetic (`a`), uppercase roman (`R`), and uppercase alphabetic (`A`). The string bound to `page_precedence` (default `"rnaRA"`) specifies their order.

Line Wrapping In the output index file, the merged list of page numbers can be wrapped in multiple lines, if it is longer than `line_max`. The newly wrapped line is indented by `indent_space` whose length is `indent_length`. This artificial line wrapping does not make any difference in formatting, but does provide increased readability for the preformatted final index. This feature may seem somewhat trivial at first glance, but if no formatters are involved whatsoever, the readability of the verbatim output index is important.

34.2 Processing

The output format is governed by an index style `.ist`. Writing new ones is a black art.

The file `gind.ist` is shown below

```
actual '='
encap '|'
level '>'
quote '!'
preamble
"\n \\begin{theindex} \n \\makeatletter\\scan@allowedfalse\n"
postamble
"\n\n \\end{theindex}\n"
item_x1   "\\efill \n \\subitem "
item_x2   "\\efill \n \\subsubitem "
delim_0   "\\pfill "
delim_1   "\\pfill "
delim_2   "\\pfill "
% The next lines will produce some warnings when
% running Makeindex as they try to cover two different
% versions of the program:
%lethead_prefix   "{\\bfseries\\hfil "
%lethead_suffix   "\\hfil}\\nopagebreak\n"
%lethead_flag       1
heading_prefix   "{\\bfseries\\hfil "
heading_suffix   "\\hfil}\\nopagebreak\n"
headings_flag       1
% Remove R so I is treated in sequence I J K not I II III
page_precedence "rnaA"
%%
%%
%% End of file `gind.ist'.
```

specifier	*attribute*	*default*	*meaning*
`preamble`	*string*	`"\\begin{theindex}\n"`	index preamble
`postamble`	*string*	`"\n\n\\end{theindex}\n"`	index postamble
`setpage_prefix`	*string*	`"\n  \\setcounter{page}{"`	page setting command prefix
`setpage_suffix`	*string*	`"}\n"`	page setting command suffix
`group_skip`	*string*	`"\n\n  \\indexspace\n"`	intergroup vertical space
`lethead_prefix`	*string*	`""`	new letter heading prefix
`lethead_suffix`	*string*	`""`	new letter heading suffix
`lethead_flag`	*number*	`0`	flag designating new letter
`item_0`	*string*	`"\n  \\item "`	level 0 item separator
`item_1`	*string*	`"\n    \\subitem "`	level 1 item separator
`item_2`	*string*	`"\n      \\subsubitem "`	level 2 item separator
`item_01`	*string*	`"\n    \\subitem "`	levels 0/1 separator
`item_x1`	*string*	`"\n    \\subitem "`	levels x/1 separator
`item_12`	*string*	`"\n      \\subsubitem "`	levels 1/2 separator
`item_x2`	*string*	`"\n      \\subsubitem "`	levels x/2 separator
`delim_0`	*string*	`", "`	level 0 key/page delimiter
`delim_1`	*string*	`", "`	level 1 key/page delimiter
`delim_2`	*string*	`", "`	level 2 key/page delimiter
`delim_n`	*string*	`", "`	inter page number delimiter
`delim_r`	*string*	`"--"`	page range designator
`encap_prefix`	*string*	`"\\"`	page encapsulator prefix
`encap_infix`	*string*	`"{"`	page encapsulator infix
`encap_suffix`	*string*	`"}".`	page encapsulator suffix
`page_precedence`	*string*	`"rnaRA"`	page type precedence
`line_max`	*number*	`72`	maximum line length
`indent_space`	*string*	`"\t\t"`	indentation for wrapped lines
`indent_length`	*number*	`16`	length of indentation

Table 34.3: Output style parameters.

34.3 Customization

When creating an index with **makeindex** one can create a `sample.ist` file that can be used together with the `makeidx` program to customize the way the index will look.

```
heading_prefix "{\\bfseries\\hfil "
heading_suffix "\\hfil}\\nopagebreak\n"
headings_flag 1
delim_0 "\\dotfill"
delim_1 "\\dotfill"
delim_2 "\\dotfill"
```

This will write the first alphabet symbol in bold font, and uses dots as delimiters. This file is generally used jointly with `makeindex` using

```
makeindex -s sample.ist filename.idx
```

where `filename.idx` has been craeted by executing `latex` or one of the other engine commands such as `pdflatex` on `filename.tex`.
According to **gabora**, you may use

```
sort_rule "\." "\b\."
```

```
sort_rule "\:" "\b\:"
sort_rule "\," "\b\,"
```

34.4 Writing custom indexing commands

For complex documents it is easier to write a number of macros to assist with indexing and to also provide consistency. For example if you want to index the Devanagari alphabet we might need to get quite creative as to how to both index it as well as get the symbols in the index.

34.5 Multiple indices

The **multind**[2] extends the `makeidx` macros to enable multiple indices. The commands are the same as those for `mkidx`, but you have to pass an extra parameter `name` as the first argument to the commands.
printindex

```
\usepackage{multind}
\makeindex{books}
\makeindex{authors}
...
\index{books}{A book to index}
\index{authors}{Put this author in the index}
...
\printindex{books}{The Books index}
\printindex{authors}{The Authors index}
```

This package has multiple issues. It is not found in normal TeX distributions and also has compatibility issues with $\mathcal{A}\mathcal{M}\mathcal{S}$ classes and packages. It is also fairly old, as it was written for LaTeX 2.09. A series of newer packages are now available.
The package **imakeidx**[3] by Enrico Gregorio

34.6 How to I add index to the table of contents?

By default LaTeX will not show in Table of Contents. It can be added manually. To add index as a chapter, use the control sequence `\addcontentsline`, as shown below:

```
\clearpage
\addcontentsline{toc}{chapter}{Index}
\printindex
```

```
\clearpage
\addcontentsline{toc}{chapter}{Index}
\printindex
```

[2] **multind**.
[3] Gregorio, Enrico (Oct. 2016). *The package imakeidx*. Version v1.3e.

34.7 Limitations of the MakeIndex index processor

The `MakeIndex` program has served the LaTeX community well for many years. With the newer engines it struggles with indexing `utf` documents and this is its major limitation. Its limitations with multiple indexing can be by-passed using the **imakeidx** package.

34.8 The xindy index processor

A more recent index processor is xindy.

xindy means fle**x**ible **ind**exing s**y**stem. It is an indexing system that can be used to generate book-like indexes for arbitrary document preparation systems. This term includes systems such as TeX and LaTeX , the Nroff-family or SGML-based systems (e.g. HTML) that process some kind of text and generate indexing information. It is not fixed to any specific system, but can be configured for a wide variety of purposes.

Authors sometimes wish to include an index into their document, but very often their document preparation systems aren't able to produce indexes on their own, or the capabilities of the built-in indexers are insufficient and produce ugly looking results. More often they use separate specialized tools for this purpose, sometimes called index processors.

In comparison to other index processors has several powerful features that make it an ideal framework for describing and generating complex indexes. Its most interesting features are

34.8.1 Internationalization

xindy can be configured to process indexes for many languages with different letter sets and different sorting rules. For example, many roman languages such as Italian, French, Portuguese or Spanish contain accentuated letters such as À, Á, ñ. Other languages from northern Europe have letters like Ä, Ø, æ or ß which often can't even be processed by many index processors let alone sorting them correctly into an index. The xindy system can be configured to process these alphabets by defining sort and merge rules that allow expressing language specific rules. One example of such a rule would be

(sort-rule "ä" "ae")

defining that a word containing the umlaut-a will be sorted as if it contained the letters ae instead. This is one form of how the umlaut-a ("ä") is sorted into german indexes. With an appropriate set of rules on can express the complete rules of a specific language.

phd package and give a background as to how fonts are used in LaTeX.

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`.

```
%<*2ekernel>
\message{index,}
```

\makeindex

Opens the jobname.idx file for writing. This must appear before the `\begin{document}` command.[1]

```
\def\makeindex{%
  \newwrite\@indexfile
  \immediate\openout\@indexfile=\jobname.idx
  \def\index{\@bsphack\begingroup
             \@sanitize
             \@wrindex}\typeout
    {Writing index file \jobname.idx}%
  \let\makeindex\@empty
}
\@onlypreamble\makeindex
```

\@wrindex

```
 \def\@wrindex#1{%
   \protected@write\@indexfile{}%
      {\string\indexentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

(*End definition for* `\@wrindex`.)

\index

```
 \def\index{\@bsphack\begingroup \@sanitize\@index}
```

(*End definition for* `\index`.)

\@index

```
\def\@index#1{\endgroup\@esphack}
```

(*End definition for* `\@index`.)

\makeglossary

```
\def\makeglossary{%
  \newwrite\@glossaryfile
  \immediate\openout\@glossaryfile=\jobname.glo
  \def\glossary{\@bsphack\begingroup
                \@sanitize
                \@wrglossary}\typeout
    {Writing glossary file \jobname.glo }%
```

[1] v1.1b,1994/11/04,Removed `\if@filesw` from `\makeindex`.

[2] Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```
  \let\makeglossary\@empty
}
\@onlypreamble\makeglossary
```

(*End definition for* `\makeglossary`.)

`\@wrglossary`

```
\def\@wrglossary#1{%
   \protected@write\@glossaryfile{}%
      {\string\glossaryentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

(*End definition for* `\@wrglossary`.)

`\glossary`

```
\def\glossary{\@bsphack\begingroup\@sanitize\@index}
```

(*End definition for* `\glossary`.)

```
%</2ekernel>
```

[2] v1.1e,1996/01/20,Make no-op after use pr/2048.

phd package and give a background as to how fonts are used in LaTeX.

36.1 Introduction

This section of the kernel deals with citations and bibliographies. With the exception of some journals and short papers and packages, it is unlikely that some package such as **natbib** or **biblatex** will be loaded.

The class was wrritten and maintained by Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Chris Rowley, Rainer Schöpf.

36.2 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{⟨*file*$_1$, *file*$_2$, *file*$_3$, ..., *file*$_n$⟩}

\bibliography{⟨*file1,file2, ...,filen*⟩} : specifies the bibdata files. Writes a **\bibdata** entry on the **.aux** file and tries to read in **mainfile.bbl**.

\bibliographystyle **\bibliographystyle**{⟨*style*⟩} : Writes a **\bibstyle** entry on the **.aux** file.

thebibliography The **thebibliography** environment is a list environment. To save the use of an extra counter, it should use **enumiv** as the item counter. Instead of using **\item**, items in the bibliography are produced by the following commands:
\bibitem{⟨*name*⟩} : Produces a numbered entry cited as ⟨*name*⟩.
\bibitem[⟨*label*⟩]{⟨*name*⟩} : Produces an entry labeled by ⟨*Label*⟩ and cited by ⟨*name*⟩.

The former is used for bibliographies with citations like [1], [2], etc.; the latter is used for citations like [Knuth82].

The document class must define the thebibliography environment. This environment has a single argument, which is the widest bibliography label– e.g., if the [Knuth67] is the widest entry, then this argument will be Knuth67. The **\thebibliography** command must begin a list environment, which the **\endthebibliography** command ends.

\cite Entries are cited by the command **\cite**{⟨*name*⟩}.

\nocite **\nocite**{⟨*citations*⟩} puts information on the **.aux** file that causes BIBTEX to include the {⟨*citations*⟩} list in the bibliography, but puts nothing in the text.

\nocite{*} is special: it tells BIBTEX to put the whole of a collection of references into the bibiography.

```

\message{bibliography,}
 PARAMETERS

  \@cite   : A macro such that \@cite{LABEL1,LABEL2}{NOTE}
             produces the output for a ↩
                  ↪\cite[NOTE]{FOO1,FOO2} command,
             where entry FOOi is defined by ↩
                  ↪\bibitem[LABELi]{FOOi}.
```

```
              The switch @tempswa is true if the optional NOTE argument
              is present.
              The default definition is :
                \@cite{LABELS}{NOTE} ==
                   BEGIN [LABELS
                         IF @tempswa = T THEN , NOTE FI
                         ]
                   END

   \@biblabel : A macro to produce the label in the bibliography
                entry.  For \bibitem[LABEL]{NAME}, the label is
                generated by \@biblabel{LABEL}.  It has the default
                definition \@biblabel{LABEL} -> [LABEL].
  CONVENTION

  \b@FOO : The name or number of the reference created by \cite{FOO}
           E.g., if \cite{FOO} -> [17] , then \b@FOO -> 17.
```

\bibitem

```
\def\bibitem{\@ifnextchar[\@lbibitem\@bibitem}
```

\@lbibitem

```
\def\@lbibitem[#1]#2{\item[\@biblabel{#1}\hfill]\if@filesw
      {\let\protect\noexpand
       \immediate
       \write\@auxout{\string\bibcite{#2}{#1}}}\fi\ignorespaces}
```

\@bibitem

```
\def\@bibitem#1{\item\if@filesw \immediate\write\@auxout
       {\string\bibcite{#1}{\the\value{\@listctr}}}\fi\ignorespaces}
```

\bibcite

```
\def\bibcite{\@newl@bel b}
```

\citation

```
\let\citation\@gobble
```

\cite

```
\DeclareRobustCommand\cite{%
  \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}}
```

\@citex

```
\def\@citex[#1]#2{\leavevmode
  \let\@citea\@empty
  \@cite{\@for\@citeb:=#2\do
    {\@citea\def\@citea{,\penalty\@m\ }%
     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
```

Using `\hbox` instead of `\mbox` is fine because of the `\leavevmode` above. In fact the use of a box around the citation contents is more than questionable in my view (FMi), but within 2e I have to keep that for compatibility reasons as it would probably change too many existing documents. Its main reason is to avoid hyphenation of labels such as [FOOB89] into [FOO-B89] so in certain styles it makes sense; but, for example, in author year citations it becomes more than questionable.
So Chris added yet another hook here, as suggested by, at least, Donald Arsenau. Note that this one is inside the first argument of the `\@cite` hook. This decouples the top-level typesetting of the citation from the details of the other business conducted here. All this really needs a complete rethink to get the right modularity.

```
     \@ifundefined{b@\@citeb}{\hbox{\reset@font\bfseries ?}%
       \G@refundefinedtrue
       \@latex@warning
         {Citation `\@citeb' on page \thepage \space ↩
              ↪undefined}}%
       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}}{#1}}
```

\bibdata

\bibstyle

```
\let\bibdata=\@gobble
\let\bibstyle=\@gobble
```

\bibliography

```
\def\bibliography#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibdata{#1}}%
  \fi
  \@input@{\jobname.bbl}}
```

\bibliographystyle

```
\def\bibliographystyle#1{%
  \ifx\@begindocumenthook\@undefined\else
    \expandafter\AtBeginDocument
  \fi
    {\if@filesw
       \immediate\write\@auxout{\string\bibstyle{#1}}%
     \fi}}
```

\nocite

```
\def\nocite#1{\@bsphack
```

```
  \ifx\@onlypreamble\document
    \@for\@citeb:=#1\do{%
      \edef\@citeb{\expandafter\@firstofone\@citeb}%
      \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
      \@ifundefined{b@\@citeb}{\G@refundefinedtrue
          \@latex@warning{Citation `\@citeb' undefined}}{}}%
  \else
    \@latex@error{Cannot be used in preamble}\@eha
    % \AtBeginDocument{\nocite{#1}}
  \fi
  \@esphack}
\expandafter\let\csname b@*\endcsname\@empty
```

36.2.1 Default definitions

This hook determines the 'relative formatting' of the two logical parts of a citation with comment.

\@cite

```
\def\@cite#1#2{[{#1\if@tempswa , #2\fi}]}
```

\@cite@ofmt

```
\let\@cite@ofmt\hbox
```

\@biblabel

```
\def\@biblabel#1{[#1]}
```

phd package and give a background as to how fonts are used in LaTeX.

37.1 Introduction

This chapter explains how the LaTeX kernel defines commands and styles for running headings. It is mostly an extract from the kernel listing with some additional explanations. It can be found in the file `ltpage.dtx`. It also deals with other commands and environments that can affect the final layout of a page. Some of these commands are only used by the output routine.

37.2 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.

37.3 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.)

37.4 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 oc-

cur on the same page.

Commands like **\tableofcontents** that should set the marks in some page styles use a **\@mkboth** command, which is **\let** by the pagestyle command (**\ps@...**) to **\markboth** for setting the heading or to **\@gobbletwo** to do nothing.

```
%<*2ekernel>
```

\pagestyle{⟨*page style name*⟩}

User command to set the page style for this and following pages.

```
\def\pagestyle#1{%
  \@ifundefined{ps@#1}%
    \undefinedpagestyle
    {\@nameuse{ps@#1}}}
```

\thispagestyle User command to set the page style for this page only.

```
\def\thispagestyle#1{%
  \@ifundefined{ps@#1}%
    \undefinedpagestyle
    {\global\@specialpagetrue\gdef\@specialstyle{#1}}}
```

(*End definition for* `\thispagestyle`.)

\ps@empty

The empty page style: No head or foot line.

```
\def\ps@empty{%
  \let\@mkboth\@gobbletwo\let\@oddhead\@empty\let\@oddfoot\@empty
  \let\@evenhead\@empty\let\@evenfoot\@empty}
```

\ps@plain

The plain page style: No head, centred page number in foot.

```
\def\ps@plain{\let\@mkboth\@gobbletwo
     \let\@oddhead\@empty\def\@oddfoot{\reset@font\hfil\thepage
     \hfil}\let\@evenhead\@empty\let\@evenfoot\@oddfoot}
```

\@leftmark **\@rightmark** We implement **\@leftmark** and **\@rightmark** in terms of already defined commands to save token space. We can't get rid of them since they are sometimes used in applications.

```
\let\@leftmark\@firstoftwo
\let\@rightmark\@secondoftwo
```

(*End definition for* `\@leftmark` *and* `\@rightmark`.)

\markboth **\markright** User commands for setting LaTeX marks.

```
\def\markboth#1#2{%
  \begingroup
    \let\label\relax \let\index\relax \let\glossary\relax
    \unrestored@protected@xdef\@themark {{#1}{#2}}%
    \@temptokena \expandafter{\@themark}%
    \mark{\the\@temptokena}%
```

```
  \endgroup
  \if@nobreak\ifvmode\nobreak\fi\fi}
\def\markright#1{%
  \begingroup
    \let\label\relax \let\index\relax \let\glossary\relax
    \expandafter\@markright\@themark {#1}%
    \@temptokena \expandafter{\@themark}%
    \mark{\the\@temptokena}%
  \endgroup
  \if@nobreak\ifvmode\nobreak\fi\fi}
```

(*End definition for* `\markboth` *and* `\markright`.)

`\@markright`
`\leftmark`
`\rightmark`

```
\def\@markright#1#2#3{\@temptokena {#1}%
  \unrestored@protected@xdef\@themark{{\the\@temptokena}{#3}}}
\def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
\def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}
```

(*End definition for* `\@markright`, `\leftmark`, *and* `\rightmark`.)

`\@themark` Initialise LaTeX's marks without setting a TeX mark ⟨*whatsit*⟩.

```
\def\@themark{{}{}}
```

(*End definition for* `\@themark`.)

`\mark` Test versions of LaTeX 2_ε initialised TeX's `\mark` system at this point, but this was removed before the first release. [1] [2]

```
\AtBeginDocument{\mark{{}{}}}
```

(*End definition for* `\mark`.)

`\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 **??**[→ P. ??].) See also **??**[→ P. ??] deep in the output routine.

```
\def\raggedbottom{%
  \def\@textbottom{\vskip \z@ ﹾ
       ﹿ\@plus.0001fil}\let\@texttop\relax}
```

`\flushbottom`

Inverse of `\raggedbottom`. Makes all pages the same height.

```
\def\flushbottom{%
  \let\@textbottom\relax \let\@texttop\relax}
```

[1] LaTeX2e,1993/12/16,Init `\mark` at begin document.
[2] LaTeX2e,1993/12/17,Removed init `\mark` at begin document, since it doesn't work..

\sloppy

`\sloppy` will never (well, hardly ever) produce overfull boxes, but may produce underfull ones. (14 June 85)

```
\def\sloppy{%
  \tolerance 9999%
  \emergencystretch 3em%
  \hfuzz .5\p@
  \vfuzz\hfuzz}
```

May be this environment does not belong here, but here it is. I would have placed it with `\par` changing environments, perhaps with the lists module.

sloppypar A sloppypar environment is equivalent to `{\par \sloppy ... \par}`.

```
\def\sloppypar{\par\sloppy}
\def\endsloppypar{\par}
```

Example 4.1 Sloppy Paragraph

```
\begin{sloppypar}
\hsize=2in
\alicei
\end{sloppypar}

\bgroup
\hsize2in
\hrule
\alicei\par
\egroup
```

The King and Queen of Hearts were seated on their throne when they arrived, with a great crowd assembled about them — all sorts of little birds and beasts, as well as the whole pack of cards: the Knave was standing before them, in chains, with a soldier on each side to guard him; and near the King was the White Rabbit, with a trumpet in one hand, and a scroll of parchment in the other. In the very middle of the court was a table, with a large dish of tarts upon it: they looked so good, that it made Alice quite hungry to look at them — "I wish they'd get the trial done," she thought, "and hand round the refreshments!". But there seemed to be no chance of this, so she began looking at everything about her to

pass away the time.

The King and Queen of Hearts were seated on their throne when they arrived, with a great crowd assembled about them — all sorts of little birds and beasts, as well as the whole pack of cards: the Knave was standing before them, in chains, with a soldier on each side to guard him; and near the King was the White Rabbit, with a trumpet in one hand, and a scroll of parchment in the other. In the very middle of the court was a table, with a large dish of tarts upon it: they looked so good, that it made Alice quite hungry to look at them — "I wish they'd get the trial done," she thought, "and hand round the refreshments!". But there seemed to be no chance of this, so she began looking at everything about her to pass away the time.

`\fussy`

Resets TEX's parameters to their normal finicky values.

```
\def\fussy{%
  \emergencystretch\z@
  \tolerance 200%
  \hfuzz .1\p@
  \vfuzz\hfuzz}
```

`\overfullrule`{⟨*dim*⟩}

LATEX default is no overfull box rule. Changed by document class option.

```
\overfullrule 0pt
%</2ekernel>
```

phd package and give a background as to how fonts are used in LaTeX.

In this chapter we discuss one of the most mysterious aspects of TEX the output routine.

Sherlock Holmes in "The sign of four": "'My mind," he said, "rebels at stagnation. Give me problems, give me work, give me the most abstruse cryptogram or the most intricate analysis, and I am in my own proper atmosphere.'"

An output routine, most often now abbreviated as OR is the responsible part of the TEX algorithm to divide the document material in pages.

The output routine is considered as one of the more mysterious pieces of TEX. David Salomon wrote[1] that advanced users hardly need to be convinced that an understanding of ORs is important, since they must be used whenever, special output is desired. The chapter of the TEXbook discussing output routines claims that designing output routines makes one: ...achieve the level of a 'TEX Grandmaster'. As is so often the case, mastery of the concept of an output routine in plain TEX will only barely prepare you for the complexities awaiting you with LATEX's variant of an output routine.

The subject is considered complex for the following reasons:

1. ORs are asynchronous with the rest of TEX (this is explained later) and involve difficult concepts such as splitting boxes and insertions.
2. Certain features, which could be useful in ORs are not supported by TEX. Specifically there are no commands to identify marks, rules and `whatsits` in a box and to break up a line of text into individual characters.

TEX 's page breaking algorithm is simpler than the line breaking one. The reason for this is that global optimization of page breakpoints, the way is done in the paragraph algorithm is prohibitively in terms of memory (especially in the 1980s).

Theoretically, page breaking is a more complicated [2]than line breaking. First we will briefly discuss the algorithms that TEX actually uses.

38.1 Page breaking algorithm

The problem of page breaking has two components. One is that of stretching or shrinking available glue (mostly around display math or section headings) to find typographically desirable breakpoints. The other is that of placing 'floating' material, such as tables and figures. These are typically placed at the top or the bottom of a page, on or after the first page where they are referenced. These 'inserts', as they are called in TEX, considerably complicate the page breaking algorithms, as well as the theory.

First we start with some definitions:

[1] TUGboat/tb-11-1/tb27salomon.pdf

[2] http://www.cs.utk.edu/ eijkhout/594-LaTeX/handouts/breaking/page-tutorial.pdf

1. The *main vertical list* is where TEX collects the material that has not yet been printed. The material on the vertical list is logically divided into two parts:

   (a) The *current page*. This is the material TEX intends to print on the current page.

   (b) The *recent contributions*. This is the material, that will be committed to the current page at a later stage.

38.1.1 Typographical constraints

There are various typographical guidelines for what a page should look like, and TEX has mechanisms that can encourage, if not always enforce, this behaviour.

1. The first line of every page should be at the same distance from the top. This changes if the page starts with a section heading which is a larger type size.

2. The last line should also be at the same distance, this time from the bottom. This is easy to satisfy if all pages only contain text, but it becomes harder if there are figures, headings, and display math on the page. In that case, a 'ragged bottom' can be specified.

3. A page may absolutely not be broken between a section heading and the subsequent paragraph or subsection heading.

4. It is also desirable that:

   (a) the top of the page does not have the last line of a paragraph started on the preceding page

   (b) the bottom of the page does not have the first line of a paragraph that continues on the next page.

For ordinary purposes you will probably find that TEX's automatic method of page breaking is satisfactory. And when it occasionally gives unpleasant results, you can force the machine to break at your favorite place by typing `\eject`. But be careful: `eject` will cause TEX to stretch the page out, if necessary, so that the top and bottom baselines agree with those on other pages. If you want to eject a short page, filling it with blank space at the bottom, type `\vfill\eject` instead.

38.2 The current page and the recent contributions list

The main vertical list of TEX is divided in two parts: the *current page* and the list of *recent contributions*. Any material that is added to the main vertical list is appended to the recent contributions; the act of moving the recent contributions to the current page is known as *exercising the page builder*.

Every time something is moved to the current page, TEX computes the cost of breaking the page at that point. If it decides that it is past the optimal point, the current page up to the best break so far is put in `box255` and the remainder of the current page is moved back on top of the recent contributions. If the page is broken at a penalty, that value is recorded in `outputpenalty`, and a penalty of size 10 000 is placed on top of the recent contributions; otherwise, `outputpenalty` is set to 10 000.

If the current page is empty, discardable items that are moved from the recent contributions are discarded. This is the mechanism that lets glue disappear after a page break and at the top of the first page. When the first non-discardable item is moved to the current page, the `topskip` glue is inserted;

38.3 When is the page builder activated?

The page builder comes into play in the following circumstances.

1. Around paragraphs: after the `\everypar` tokens have been inserted, and after the paragraph has been added to the vertical list. See the end of this chapter for an example.
2. Around display formulas: after the `\everydisplay` tokens have been inserted, and after the display has been added to the list.
3. After `\par` commands, boxes, insertions, and explicit penalties in vertical mode.
4. After an output routine has ended.

In these places the page builder moves the recent contributions to the current page. Note that TEX need not be in vertical mode when the page builder is exercised. In horizontal mode, activating the page builder serves to move preceding vertical glue (for example, `\parskip`, `\abovedisplayskip`) to the page.

The `\end` command – which is only allowed in external vertical mode – terminates a TEX job, but only if the main vertical list is empty and `\deadcycles` = 0. If this is not the case the combination

```
\hbox{}\vfill\penalty+ $-2^{30}$
```

is appended, which forces the output routine to act.

38.4 The depth of the current page

The depth of the page is important since normally in good typesetting successive pages should have the same (or almost the same vertical size. (flushbottom). The height of a page is controlled and set exactly by TEX equal to `\vsize`. Consider a large `vbox` with lines of text, glue and penalties. The depth of this box, is the depth of the last component [80]. If the last component is a glue or penalty, the depth is zero. If it is a box, then its depth becomes the depth of the entire `\vbox`, except that it is limited to the value of parameter `\boxmaxdepth`.

If `\boxmaxdepth=1pt` and the depth of the bottom box is 1.94444pt, then the depth of the entire `\vbox` will be 1pt and its height will be incremented by .94444pt. This is equivalent to lowering the reference point (or,

equivalently, the baseline) of the **\vbox** by .94444pt. In the plain format, **\boxmaxdepth=\maxdimen** [348], so it has no effect on the depths of boxes. However, **\boxmaxdepth** can always be changed by the user [3]

If the last line on a page, contains letters that happen to not have any depth, the page depth will be zero. Try for example this:

```
....
\showthe\pagedepth
\bye
```

You can also try it with a LaTeX minimal and will produce the same output.

38.5 The height of a box of text

Following the literature we denote the value of **\baselineskip** (which is normally 12pt) by b. A large **\vbox** with text consists mainly of lines of text, each an **\hbox**, separated by globs of glue, normally in the (varying) amounts necessary to separate baselines by exactly b, but sometimes just the amount **\lineskip**. We assume a simple case where no large characters or equations are used. In such a case, all lines of text are separated by b. The height of the box is thus:

$$b(n-1) + \text{the height of the first line} \tag{38.1}$$

where n is the number of text lines. Remember that the first line is a special case and adjustments can be made using the value of **\topskip**.

38.6 The height of \box255

In the case of **\box**255, enough glue is placed above the first line of text to reach to **\topskip** from the first baseline. We denote the value of **\topskip** by h (10pt in plain). So if the baseline of the first line is now h below the top of the page, the height H of **\box255** should be b(n - 1) + h (Fig. 3). However, the height of **\box255** is always set, by the page builder, to **\vsize**. The difference between the two heights is usually supplied by the flexible glues on the page, the most common of which is **\parskip**

38.6.1 Dead cycles.

An execution of the OTR without shipping any material is called a *dead cycle*. Dead cycles, have their uses and we will explain this a bit later on. However, long iterations that just return *dead cycles* is an indication of an error somewhere. TeX counts the number of dead cycles in a counter named **\deadcycles** and stops the run if **\deadcycles >= \maxdeadcycles**. In the *plain* format **\maxdeadcycles** is set as 25 and in LaTeX as 0. **\maxdeadcycles = 100** is 200. Each time **\shipout** is invoked, it resets **\deadcycles** to zero.

[3] 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. LaTeX uses a value of 5.5pt whereas plain a value of 4pt [348].

```
If the file is not included, reset \deadcycles, so that a long list of non-included
files does not generate an `Output loop' error.
115 \deadcycles\z@
116 \@nameuse{cp@#1}%
117 \fi
118 \let\@auxout\@mainaux}
```

38.6.2 TEX's Page Number.

The page number can come from any source. Salomon provides an example where the OTR typesets a page number from a **\count** variable. This is typeset centered below the printed area.

Notice that the output macro, just passes the contents of the box to **\shipout**. This is not actually a very good method, but is shown here to illustrate a point.

Note the **\tenrm** in the preceding example. It is necessary because of the asynchronous nature of the OTR. When the OTR is invoked, TEX can be anywhere on the next page. Specifically, it could be inside a group where a different font is used. Without the **\tenrm**, that font (the current font) would be used in the otr. In the plain format, the **\count0** variable serves as the page number, and the following two macros are especially useful.

38.6.3 The \vsplit operation.

Supposed you have inserted the material required to go on a page on a big **\vbox**, but the material is a bit extra that what is required to fill a page exactly. You would need an operation to split the box in two. The **vsplit** operation does that. It is important to the understanding of OTR operations to have an intimate knowledge of **\vsplit**. Its syntax is:

\vsplit⟨*box number*⟩ to ⟨*dim*⟩

The result of the operation is a box. Most often it appears in an assignment such as:

```
\setbox1=\vsplit0 to2.6in
```

This sets **\box1** to a height of 2.6in, moves material from the top of **\box0** to **\box1**, and keeps the remainder in **\box0**.

\loremlines

It is important to remember that most of TEX's commands work with LATEX as well. In Example **??**, we define a box to hold **lipsum** text in a two column layout. We want to define a macro that can split the box in as many lines as we require.

(*End definition for* \loremlines.)

Example 6.1 Splitting a vbox

```
\newbox\one
\newbox\two
\long\gdef\loremlines#1#2{%
   \setbox\one=\vbox {#2}
   \setbox\two=\vsplit\one to #1\baselineskip
   \unvbox\two
   \gdef\boxone{#2}
}
\begin{multicols}{2}
\small
\loremlines{16}{\onepar}
\end{multicols}
\boxone

\setbox\one=\vbox{100}
\the\ht\one \\
\the\baselineskip
\the\splittopskip
```

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on a high and caught it, and this ball was her favorite play-

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on a high and caught it, and this ball was her favorite plaything.
4.85596pt
11.0pt10.0pt

TEX assumes that the new **\box1** may have to be shipped out as part of the page. It therefore places a glue similar to *h* at the top of **\box1**. This glue is called **\splittopskip** and has a plain format value of 10pt [348].

One important thing to note is that a box can only be split *between* lines of text. If we split a box to another size, **\box1** will come out underfull.

Here is an OTR which splits the page, ships out the top part and returns the rest to the MVL (actually, to the recent contributions):

```
\output={\setbox0=\vsplit255 to1in
\shipout\box0 \unvbox255}
```

38.7 Communicating with the OTR: Marks

The user can pass information to the output routine through *marks*. Marks have the syntax

```
\mark{mark text}
```

which is put in a mark item on the current vertical list. The mark text is subject to expansion as in **\edef**. If the mark is given in horizontal mode it migrates to the surrounding vertical lists like an insertion item (see page Text By Topic 77); however, if this is not the external vertical list, the output routine will not find the mark.

Marks are the main mechanism through which the output routine can obtain information about the contents of the currently broken-off page, in particular its top and bottom. TEX sets three variables:

\botmark the last mark occurring on the current page;

\firstmark the first mark occurring on the current page;

\topmark the last mark of the previous page, that is, the value of **\botmark** on the previous page.

If no marks have occurred yet, all three are empty; if no marks occured on the current page, all three variables are equal to the **\botmark** of the previous page.

Marks can be used to get a section heading into the headline or footline of the page.

```
\def\section#1{ ... \mark{#1} ... }
\def\rightheadline{\hbox to \hsize
    {\headlinefont \botmark\hfil\pagenumber}}
\def\leftheadline{\hbox to \hsize
   {\headlinefont \pagenumber\hfil\firstmark}}
```

This places the title of the first section that starts on a left page in the left headline, and the title of the last section that starts on the right page in the right headline. Placing the headlines on the page is the job of the output routine; see below.

It is important that no page breaks can occur in between the mark and the box that places the title:

```
\def\section#1{ ...
  \penalty\beforesectionpenalty
  \mark{#1}
  \hbox{ ... #1 ...}
  \nobreak
  \vskip\aftersectionskip
  \noindent}
```

However, useful these marks, sometimes an output routine (such as those found in LaTeX needs to know why it was invoked. Knuth discusses in the another method that involves the value of the **\outputpenalty**. By testing for this value, it is possible to see what penalty occurred at a breakpoint; any penalty of −10000, −10001, −10002, or less, forces the output routine

TEXbook: 396

to act, hence different penalty values can be used to pass different messages. (When the output routine puts material back on the list of contributions, it need not restore the penalty at the breakpoint.) If output has been forced by a highly negative value of `\outputpenalty`, the output routine can use `\vbox{\unvcopy255}` to discover how full the page-so-far actually is. Underfull and overfull boxes are not reported when `\box255` is packaged for use by the output routine, so there's no harm in ejecting a page prematurely if you want to pass a signal. (Set `\holdinginserts` positive to pass a signal when the contents of `\box255` will be sent back through the page builder again, if any insertions are present.)

Knuth also suggested another method that he called the *dirtiest trick of all* that uses the depth of `\box255`.

38.8 Insertions

Insertions are considered one of the most complex topics in TEX. Many users master topics such as tokens, file I/O, macros, and even OTRS before they dare tackle insertions. The reason is that insertions are complex, and The TEXbook, while covering all the relevant material, is somewhat cryptic regarding insertions, and lacks simple examples. The main discussion of insertions takes place on [115-1251. where TEX' s registers are also discussed. Examples of insertions are shown, mostly without explanations, on [363-364, 423-424]. A lot of what is described here is based on an article in TUGboat by David Salomon[4]

Many users understand the idea of floats. Certain material to be typeset needs to be held in a buffer and inserted at different points on a page, for example a a figure that does not fit on a page it has to be inserted at the top of the next page. An *insertion* is just a piece of a document that is generated at a certain point but appears at another point. Common examples are figures, footnotes and endnotes. Quoting Knuth:

> This algorithm is admittedly complicated, but no simpler mechanism seems to do nearly as much.

38.9 \shipout

The primitive control sequence `\shipout` is TEX's end game. It's syntax is quite simple:

```
\shipout<box>
```

From TeXbook, Chapter 23: Output Routines, page 254:

> TeX's primitive command `\shipout<box>` is what actually causes output. It sends the contents of the box to the dvi file, which is TeX's main output file; after TeX has finished, the dvi file will contain a compact device-independent encoding of instructions that specify exactly what should be printed. When a box is shipped out, TeX displays the values of `\count0` through `\count9`

[4] http://www.tug.org/TUGboat/Articles/tb11-4/tb30salomon.pdf

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Sample: This is a Test for shipout

on your terminal, as explained in Chapter 15; these ten counters are also recorded in the dvi file, where they can be used to identify the page. All of the **\openout, \closeout**, and **\write** commands that appear inside of the box are performed in their natural order as that box is being shipped out. Since a **\write** command expands macros, as explained in Chapter 21, TeX's scanning mechanism might detect syntax errors while a **\shipout** is in progress. If **\tracingoutput** is nonzero at the time of a **\shipout**, the contents of the box being shipped are written into your log file in symbolic form. You can say **\shipout** anywhere, not only in an output routine.

We can say:

```
\shipout\vbox{%
  \hrule
  \medskip
  \lipsum[1-5]
  \medskip

  This is a Test for shipout

  \hrule
}
```

Since the output box handled by TEX still holds material the page is shown in the previous page. There is no page numbers or headers and it just shows he lorem-ipsum text and a primitive caption at the bottom. I have written the example to show the difference between the logical and actual pages. TeX does not care how the page will look, it will assemble it put headers, page numbers and pass it on to shipout. Shipout will then insert it to the dvi file, which will hold all teh instructions to print a real page.

We can modify the example to add our head and foot.

```
Example 9.2 Shipout

This will print by the normal routine

\long\gdef\boxit#1#2{\hbox{\vrule
    \vbox{\hrule\kern#2pt\hbox{%
\kern#2pt\vbox{#1}\kern#2pt}\kern#2pt\hrule}\vrule}}

\makeatletter
\shipout\vbox {%
  \vskip\topsep\relax
  \vskip\headsep
   \@thehead
   \vskip30pt
    \boxit{
      \lipsum[1-5]
      }{2}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

```
   \vskip30pt
   \@thefoot
}
\makeatother
```

LaTeX's output routine takes care of all the page geometry, the details to set up the headers and footers, but most importantly intercepts teh contents of the output box measure it, cuts it inserts the inserts such as footnotes and figures, margin notes, separates the text in two columns if necessary and so on.

An output routine will prepare the virtual page and pass it onto a

Here is an OTR for a *framed* page. It surrounds the page with double rules on all sides, and centers the page number below the double box. Note that the page shipped out is wider and taller than **\box255**. The value of **\hsize** in this case is, therefore, not the width of the final page shipped out, but the width of the text lines in **\box255**.

Macro **\frameit** typesets text and surrounds it with 4 rules (see [Ex. 21.3]). Parameter #2 is the space between the rules and the text. #1 is a box containing the text.

```
Example 9.3 Example of simple output routine

\def\frameit#1#2{%
 \vbox{\hrule
  \hbox{%
    \vrule \kern#2pt
      \vbox{\kern#2pt #1
         \kern#2pt}%
      \kern#2pt\vrule}
\hrule}}

\output={
   \shipout\vbox{
   \boxit{\frameit{\box255}9}
      \medskip
      \centerline{Test Framed Page}}
  \advancepageno}
```

So far we did not care if the height of the page is right or not. In production code the shipout holds a box, which has been produced by TeX. Any material we add to it, must not affect the dimensions of the box. If we do and is too big the drivers will probably clip it.

Plain TeX has an output routine that takes care of simple things like page numbering and insertions using **\footnote** and **\topinsert**.

So far we have examined the TeX OTR in detail. I hope it has given you enough understanding, not only to write your own output routine, but also to now be ready to study the LaTeX output routine, which is much more complicated. We have so far seen that when TeX is typesetting pages of continuous text, it will gather material until it can find a least-cost page break intended to make the gathered material fit the **\pagegoal** size. The gathered material will then be placed into **\box255** and the output routine stored in the token register

`\output` will be processed in a group of its own.

Usually it will arrange the gathered material in some way, add headers, footlines and page numbers, and ship the gathered results out in typeset form with the `\shipout` command. At the time of the `\shipout` command all `\open` and `\write` commands stored in the box shipped out are expanded and written out. This is what makes it possible to have page labels corresponding to the actual page numbers at the time of shipout: the corresponding info is written to the `.aux` file at that time. The output routine may decide to place material back on the main vertical list instead of shipping it out. Its ob is to check if it can have a break, if it can it will ship the page out. If it cannot it will palce the material back on the main vertical list.

38.10 LaTeX output routines

LaTeX output routine is described in `ltoutput.dtx`. You should also have a look at `ltfloat.dtx`. The algorithm is revisited i LaTeX 3 and Frank Mittelbach, published a paper [5] in which he explains some of the problems facing the team, when dealing with the output routine.

Information on the output routine is rather scarce. Best source is a series of articles in the TUGBoat by David Salomon.

Output Routines: Examples and Techniques. Part I: Introduction and Examples.

Output Routines: Examples and Techniques. Part II: OTR Techniques

Output Routines: Examples and Techniques. Part III: Insertions

Output routines: Examples and techniques Part IV: Horizontal techniques

David Kastrup's article Output Routine Requirements for Advanced Typesetting Tasks (Proceedings of EuroTEX 2003) otlined some of the difficult areas and specifications for generic routines

The standard blocks are well described above and most tasks could be accomplished by rather working from standard building blocks like *insertion lists*, *here points*, default mechanisms for *margin notes* and so on.

38.11 Packages

OTR routines are notoriously difficult to debug and define. Some of the available packages at CTAN can make the programming job easier.

The **everypage** package by Sergio Callegari provides hooks into the LaTeX internal commands to to do actions on every page or on the current page. Specifically, actions are performed *before* the page is shipped, so they can be used to put watermarks *in the background* of a page, or to set the page layout.

The package provides two hooks:

```
\AddEverypageHook{Test}
\AddThisPageHook
```

The package reminds in some sense **bobhook**

by Karsten Tinnefeld, but it differs in the way in which the hooks are implemented, as detailed in the following. In some sense it may also be related to the

[5] http://www.latex-project.org/papers/xo-pfloat.pdf

package **everyshi**

by Martin Schroeder, but again the implementation is different.

This program adds two LaTeX hooks that get run when document pages are finalized and output to the `.dvi` or `.pdf` file. Specifically, one hook gets executed on every page, while the other is executed for the current page. Hook actions are are performed *before* the page is output on the medium, and this is important to be able to play with the page layout or to put things *behind* the page contents (e.g., watermarks such as an image, framing, the "DRAFT" word, and the like).

The package reminds in some sense **bobhook** by Karsten Tinnefeld, but it differs in the way in which the hooks are implemented:

1. there is no formatting inherent in the hooks. If one wants to put some watermark on a page, it is his own duty to put in the hook the code to place the watermark in the right position. Also note that the hooks code should *eat up no space* in the page. Again, if the hooks are meant to place some material on the page, it is the duty of the hook programmer to put code in the hooks to pretend that the material has zero width and zero height. The implementation is *lighter* than the **bobhook**

   one, and possibly more flexible, since one is not limited by any pre-coded formatting for the hooks. On the other hand it is possibly more difficult to use. Nonetheless, it is easy to think of other packages relying on **everypage**

   to deliver more user-friendly and *task specific* interfaces. Already there are a couple of them: the package **flippdf**

   produces mirrored pages in a PDF document and **draftwatermark**

   watermarks document pages.

2. similarly to **bobhook**

   and **watermark**

   , the package relies on the manipolatoin of the internal LaTeX macro `\@begindvi` to do the job. However, the redefinition of `\@begindvi` is here postponed as much as possible, striving to avoid interference with other packages using `\AtBeginDvi` or anyway manipulating `\@begindvi`. Specifically **everypage**

   makes no special assumption on the initial code that `\@begindvi` might contain.

Also in some sense **everypage**

can be related to package **everyshi**

by Martin Schröeder **everyshi**, but it differs radically from it in the implementation. In fact,**everypage**

operates by manipulation of the `\@begindvi` macro, rather than at the lower level `\shipout` macro.

38.12 hooking at shipout

`\EveryShipout`

\AtNextShipout

This package provides the hooks **\EveryShipout** and **\AtNextShipout** whose arguments are executed after the output routine has constructed **\box255**, and before **\shipout** is called.

An example application for this package would be a package for adding text to the bottom of each page. The **prelim2e** package adopts this method prelim2e.[6]

The solution uses is based on code developed in **quire.tex** by Marcel R. van der Goot.[7]

The **prelim2e** intercepts and modifies the **\box255**.

```
44 \newcommand{\@Prelim@EveryShipout}{%
45 \bgroup
% First we save the dimensions of \box255: height, width and depth; and calculate
% the total height of \box255.
46 \dimen\z@=\wd\@cclv
47 \dimen\@ne=\ht\@cclv
48 \dimen\tw@=\dp\@cclv
49 \dimen\thr@@=\dimen1
50 \advance\dimen\thr@@ by \dimen\tw@
% Then we set \box255: A \vbox to the total height of \box255. In this a \hbox to
% the width of \box255 is included, in which \box255 is set.
51 \global\setbox\@cclv\vbox to \dimen\thr@@{%
52 \hb@xt@\dimen\z@{%
53 \box\@cclv%
54 \hss
55 }%
```

To this we append the text produced by **\PrelimText**. It is put in a **\vbox to 0pt** in which a **\hbox** to the width of **\box255** is included, in which **\PrelimText** is set. We have to reset **\protect** because it is set to **\noexpand** by the output routine.

```
56 \vbox to \z@{%
57   \hb@xt@\dimen\z@{%
58     \let\protect\relax
59     \hfill\PrelimText\hfill
60   }%
61   \vss
62 }%
63   \vss
64 }%
```

Finally we set the dimensions of **\box255** to the values they had before **\@Prelim@EveryShipout**.

```
65 \wd\@cclv=\dimen\z@
66 \ht\@cclv=\dimen\@ne
67 \dp\@cclv=\dimen\tw@
```

[6] **prelim2e**.
[7] **quire**.

```
68 \egroup
69}
```

Once the command is defined, it is hooked into the system via `\EveryShipout` when it is in draft mode.

```
70 \if@prelim@draft
71 \EveryShipout{\@Prelim@EveryShipout}
72 \fi
```

38.13 How to place a background image

One can use Ti*k*Z to place a background image or some text on a page

First we define some utility macros:

```
\def\bg@contents{Draft}
\def\bg@color{red!45}
\def\bg@angle{60}
\def\bg@opacity{.5}
\def\bg@scale{15}
\def\bg@position{current page.center}
\def\bg@anchor{}
\def\bg@hshift{0}
\def\bg@vshift{0}
```

A new command is then developed to describe the background material

```
\newcommand\bg@material{%
   \begin{tikzpicture}[remember picture,overlay]
   \node ↩
        ↩[rotate=\bg@angle,scale=\bg@scale,opacity=\bg@opacity,%
   xshift=\bg@hshift,yshift=\bg@vshift,color=\bg@color]
   at (\bg@position) [\bg@anchor] {\bg@contents};
  \end{tikzpicture}}%
```

Once the background material has been defined we can place it on the page by simply calling:

```
\newcommand\BgThispage{\AddThispageHook{\bg@material}}
```

The **background**

[8] package by **h**as capitalized on two good packages the Ti*k*Z and the **everypage**

.[9] As most of modern TEX programming works with `pdf` files, package developers prefer to use Ti*k*Z methods for hooking directly into the pdf and thus avoid a trip into the output routine. If it is required then it hooks via the dvi or shipout commands.[10]

[8] **background**.

[9] **everypage**.

[10] These packages are loaded automatically by the **phd-pkgmanager**

.

phd package and give a background as to how fonts are used in LaTeX.

39.1 Introduction

In Chapter **??**, we described the mechanics of output routines both as found in Plain TEX and in LATEX 2_ε. This is a longer treatise of the subject and includes commentary on the actual listing as found in LATEX 2_ε. The Output Routine (OR) or (OTR) as is sometimes denoted in the literature, is the procedure by which LATEX 2_ε assembles the material that makes a page by combining text and floats, adding any other inserts such as footnotes, headers and footers and then ships out the page to produce a `.dvi` or with some TEX engines to be translated straight into `.pdf` output. It is a very complex process as it has to keep a lot of different material in different lists and boxes.

The output routine as defined in the kernel covers a lot of functionality.

1. Defines page geometry parameters.
2. Positions floats.
3. Adds headers and footers.
4. Adds hooks.

39.2 Floats

The interesting and complicated part of the OR is its algorithm for handling floats. The floating environments are defined by the standard classes. For example the `book` class defines both the figure as well as the table environments. It is instructive to start the discussion of the algorithm from this point.

The class defines the default float placement specifier using `\fps@figure` and then goes on to define the figure environment with the help of **??**$^{\to \text{P.}??}$ and **??**$^{\to \text{P.}??}$, which are defined in the generic `float.dtx` in the kernel.

```
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
```

A float specifier is made of two parts the float type, which is a power of two–e.g., figures in the case of the book class are type 1 and tables type 2 and the *placement specification* describing where the float can be placed. The type is defined in powers of two due to the way the specifiers are represented using a binary representation internally.

110010

50

So a new float will need to be shifted in powers of two. The kernel defines routines to check for various combinations. This has simplified the programming although it may not be easy to follow. The float specifier is encoded as follows, where bit 0 is the least significant bit.

	1	1	0	0	1	0	= 50
1	0	1	0	0	1	0	= 82

Figure 39.1: Binary representation of float specifiers. Top is for figure with [t] option and the bottom is for table with the [t] option.

Bit	Meaning
0	1 if the float may go where it appears in the text.
1	1 if the float may go on the top of a page.
2	1 if the float may go on the bottom of a page.
3	1 if the float may go on a float page.
4	1 unless the *placement* includes a !
5	1 if a type 1 float
6	1 if a type 2 float etc.

If a number is odd denotes a here placement specification [h] and if it is negative a marginpar. Since TeX's number limit of $2^{31} - 1$ and the first 5 bits are taken by the float identifiers there remain 26 available float types for the adventurous. http://tex.stackexchange.com/questions/32359/what-is-the-exact-purpose-of-ftypetype.

We have seen so far how a float is defined in the standard classes and what type of parameters are coded. Once LaTeX 2_ϵ sees the environment it will execute the macro `\@float` which is defined in the `float.dtx`.

Once in `\@float` the bits are set as well as the relevant penalties. The penalties are distinct in order to signal to the output routines the type of float.

One needs to remember that the floats are placed based on constraints and page sizing parameters.

39.3 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`.

39.4 Page Style Parameters

`\floatsep`	Space left between floats.
`\textfloatsep`	Space between last top float or first bottom float and the text.
`\topfigrule`	Command to place rule (or whatever) between floats at top of page and text. Executed in inner vertical mode right before the `\textfloatsep` skip separating the floats from the text. Must occupy zero vertical space. (See `\footnoterule`.)
`\botfigrule`	Same as `\topfigrule`, but put after the `\textfloatsep` skip separating text from the floats at bottom of page.
`\intextsep`	Space left on top and bottom of an in-text float.
`\dblfloatsep`	Space between double-column floats.
`\dbltextfloatsep`	Space between top double-column floats and text.
`\dblfigrule`	Similar to `\topfigrule`, but for double-column floats.
`\@fptop`	Glue to go at top of float column – must be 0pt + stretch
`\@fpsep`	Glue to go between floats in a float column.
`\@fpbot`	Glue to go at bottom of float column – must be 0pt + stretch
`\@dblfptop`, `\@dblfpsep`, `\@dblfpbot`	Analogous for double-column float page in two-column format.
`@twocolumn`	Boolean. T if two columns per page globally.
`\@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.

39.5 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. It is used locally in processing double floats.
`\@fpmin`	L Minimum height of floats in a float column.

The macro `\@dblfloatplacement` sets the following parameters.

`\@dbltopnum`	G Maximum number of double-column floats allowed at the top of a two-column page.
`\@dbltoproom`	G Maximum height of double-column floats allowed at top of two-column page.
`\@fpmin`	L Minimum height of floats in a float column.

It should also perform the following local assignments where necessary – i.e., where the new value differs from the old one:

`\@fptop`	L `\@dblfptop`
`\@fpsep`	L `\@dblfpsep`
`\@fpbot`	L `\@dblfpbot`

39.6 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`.

39.7 Calling the output routine

The output routine is called either by TeX's normal page-breaking mechanism, or by a macro putting a penalty ≤ -10000 in the output list. In the latter case, the penalty indicates why the output routine was called, using the following code.

Note that a float or marginpar puts the following sequence in the output list:

1. a penalty of -10004,
2. a null `\vbox`
3. a penalty of -10002 or -10003.

This solves two special problems:

Penalty	Reason
-10000	`\pagebreak` `\newpage`
-10001	`\clearpage` (`\penalty` -10000 `\vbox{}` `\penalty` -10001)
-10002	float insertion, called from horizontal mode
-10003	float insertion, called from vertical mode.
-10004	float insertion.

Table 39.4: Penalties when calling the output routine.

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.

39.8 Functions used in the output routine

`\@outputpage`

`\@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:

```
begin
  if \@thispagestyle = true then
    use \thispagestyle style
  end
  use ordinary page style.
end
```

`\@tryfcolumn`

`\@tryfcolumn\FLIST` tries to form a float column composed of floats from `\FLIST` (if nonempty) with the following parameters see **??**:

`\@colht`	height of box
`\@fpmin`	minimum height of floats in the box
`\@fpsep`	interfloat space
`\@fptop`	glue at top of box
`\@fpbot`	glue at bottom of box.

If it succeeds, then it does the following:

`\@outputbox`	L the composed float box.
`@fcolmade`	G true
`\FLIST`	G `\FLIST` - floats put in box
`\@freelist`	G `\@freelist` + floats put in box

If it fails, then:

`@fcolmade`	G false	NOTE: BIT MUST BE A SINGLE TOKEN!

\@makefcolumn

\@makefcolumn \FLIST is similar to \@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

\@startcolumn Calls \@tryfcolumn\@deferlist. If \@tryfcolumn returns with (globally set) @fcolmade = false, then:

Globally sets \@toplist and \@botlist to floats from \@deferlist to go at top and bottom of column, deleting them from \@deferlist. It does this using \@colht as the total height, the page style parameters \@floatsep and \@textfloatsep, and the float placement parameters \@topnum, \@toproom, \@botnum, \@botroom, \@colnum and \textfraction.

Globally sets \@colroom to \@colht minus the height of the added floats.

\@startdblcolumn

Calls \@tryfcolumn\@dbldeferlist{8}. If \@tryfcolumn returns with (globally set) @fcolmade = false, then:
* Globally sets \@dbltoplist to floats from \@dbldeferlist to go at top and bottom of column, deleting them from \@dbldeferlist. It does this using \textheight as the total height, and the parameters \@dblfloatsep, etc.
* Globally sets \@colht to \textheight minus the height of the added floats.

\@combinefloats

\@combinefloats Combines the text from box \@outputbox with the floats from \@toplist and \@botlist, putting the new box in \@outputbox. It uses \floatsep and \textfloatsep for the appropriate separations. It puts the elements of \TOPLIST and \BOTLIST onto \@freelist, and makes those lists null.

\@makecol

\@makecol Makes the contents of \box255 plus the accumulated footnotes, plus the floats in \@toplist and \@botlist, into a single column of height \@colht (unless the page height has been locally changed), which it puts into box \@outputbox. It puts boxes in \@midlist back onto \@freelist and restores \maxdepth.

\@opcol

\@opcol Outputs a column whose text is in box \@outputbox

39.9 User commands that call affect the output routine

\newpage

```
\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.
```

39.10 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 if the float may go where it appears in the text.
1	1 if the float may go on the top of a page.
2	1 if the float may go on the bottom of a page.
3	1 if the float may go on a float page.
4	1 unless the PLACEMENT incluses a !
5	if a type 1 float
6	1 if a type 2 float etc.

A negative float specifier is used to indicate a marginal note.

39.11 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. Of interest here—and a bit difficult to follow is `\@next` see **??**.

```
\@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*⟩}

`\@bitor\NUM\LIST` Globally sets switch `@test` to the disjunction for all I of bit log2 `\NUM` of the float specifiers of all the floats in `\LIST`.
I.e., @test is set to true iff there is at least one float in `\LIST` having bit log2 `\NUM` of its float specifier equal to 1.

```
%  Note: log2 [(\count I)/32] is the bit number corresponding to the
%  type of float I.  To see if there is any float in \LIST having
```

```
%  the same type as float I, you run \@bitor with
%
%    \NUM = [(\count I)/32] * 32.
%
% \@bitor\NUM\LIST ==
%   BEGIN
%      @test :=G false
%      { \@elt \CTR ==  if \NUM <> 0 then
%                          if \count\CTR / \NUM is odd
%                             then  @test := true       fi fi
%        \LIST
%      }
%   END
%
%
% \@cons\LIST\NUM : Globally sets \LIST := \LIST * \@elt \NUM
%
% \@cons\LIST\NUM ==
%   BEGIN {  \@elt == \relax
%            \LIST :=G \LIST \@elt \NUM
%         }
```

39.12 Box lists for float-placement algorithms

The **fixltx2e** the now redundant package—as it was incorporated into the fixes provided by `fixltx2e`— modify the output routine to correct a problem with synchronizing floats in double column texts with those in single column **fix2col**; **fixltx2e**. Here we will describe the normal behaviour. Additional commentary on the changes is discussed later on.

`\@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.

```
Example 12.1 Meaning toplist

\makeatletter
\meaning\@freelist\\
\meaning\@toplist
\makeatother
```

macro:->\@elt \bx@G \@elt \bx@H \@elt \bx@I \@elt \bx@J \@elt \bx@K \@elt \bx@L \@elt \bx@M \@elt \bx@N \@elt \bx@O \@elt \bx@P \@elt \bx@Q \@elt \bx@R \@elt \bx@S \@elt \bx@T \@elt \bx@V \@elt \bx@W \@elt \bx@X \@elt \bx@Y \@elt \bx@Z \@elt \bx@AA \@elt \bx@BB \@elt \bx@CC \@elt \bx@DD \@elt \bx@EE \@elt \bx@FF \@elt \bx@GG \@elt \bx@U \@elt \bx@HH \@elt \bx@II \@elt \bx@JJ \@elt \bx@KK \@elt \bx@LL \@elt \bx@MM \@elt \bx@NN \@elt \bx@OO \@elt \bx@PP \@elt \bx@QQ \@elt \bx@RR \@elt \bx@SS \@elt \bx@TT \@elt \bx@UU \@elt \bx@VV \@elt \bx@WW \@elt \bx@XX \@elt \bx@YY \@elt \bx@ZZ \@elt \bx@A \@elt \bx@B \@elt \bx@C \@elt \bx@D \@elt \bx@E \@elt \bx@F
macro:->

39.13 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 opt

\@addtotoporbot

Tries to put insert **\@currbox** on **\@toplist** or **\@botlist**.
Called only under same conditions as **\@addtobot**.
If it succeeds, then: * sets @insert true * decrements **\@toproom** or **\@botroom** by **\ht** BOX * decrements **\@colnum** and either **\@topnum**

or **\@botnum** by 1 * decrements **\@colroom** by **\ht** BOX + **\floatsep** or **\textfloatsep**, as appropriate.

\@addtocurcol

Tries to add **\@currbox** to current column, setting @insert true if it succeeds, false otherwise. It will add **\@currbox** to top only if bit 0 of **\count \@currbox** is 0, and to the bottom only if bit 0 = 0 or an earlier float of the same type is put on the bottom.
If the float is put in the text, then **\penalty\interlinepenalty** is put right after the float, before the following **\vskip**, and
\outputpenalty :=L 0.

\@addtonextcol

Tries to add **\@currbox** to the next column, setting @insert true if it succeeds, false otherwise.

\@addtodblcol

Tries to add **\@currbox** to the next double-column page, adding it to **\@dbltoplist** if it succeeds and **\@dbldeferlist** if it fails.

Floats and marginpars add a lot of dead cycles.

```
\maxdeadcycles = 100
\let\@elt\relax
```

\@next

```
\def\@next#1#2#3#4{\ifx#2\@empty #4\else
   \expandafter\@xnext #2\@@#1#2#3\fi}
```

```
\def\@xnext \@elt #1#2\@@#3#4{\def#3{#1}\gdef#4{#2}}
```

```
\def\@testfalse{\global\let\if@test\iffalse}
\def\@testtrue {\global\let\if@test\iftrue}
\@testfalse
```

[1]

```
\def\@bitor#1#2{\@testfalse {\let\@elt\@xbitor
   \@tempcnta #1\relax #2}}
```

```
\def\@xbitor #1{\@tempcntb \count#1
   \ifnum \@tempcnta =\z@
   \else
     \divide\@tempcntb\@tempcnta
     \ifodd\@tempcntb \@testtrue\fi
   \fi}
```

[1] v1.1v 1996/07/26 remove **\global** before **\@test...**

39.14 Definition of Float Boxes (inserts)

All boxes are defined using **??**[→ P. **??**]. A total of eighteen insertions are defined here and later on inserted in the freelist (See line **??**).

```
\newinsert\bx@A \newinsert\bx@B \newinsert\bx@C
\newinsert\bx@D \newinsert\bx@E \newinsert\bx@F
\newinsert\bx@G \newinsert\bx@H \newinsert\bx@I
\newinsert\bx@J \newinsert\bx@K \newinsert\bx@L
\newinsert\bx@M \newinsert\bx@N \newinsert\bx@O
\newinsert\bx@P \newinsert\bx@Q \newinsert\bx@R
```

TEX allows 255 classes of insertions `\insert0` to `\insert254`. It is important to remember that every insert is tied to other registers of the same number. For example, `\insert100` is connected with `\count100`, `\dimen100`, `\skip100` and `\box100`. plain TEX provides an allocation function for insertions as it does for registers. Appendix B includes the command:

```
\newinsert\footins
```

which defines `\footins` as the number for footnote insertions.

LATEX 2e adopts similar definitions (see Chapter **??**). In the latest versions allocations are extended with `\extrafloats`.

The `\@freelist` is defined next. Notice that `\@elt` is included here to enable the manipulation of the list later on.

```
\gdef\@freelist{\@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
                \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
                 \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
                 \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
```

The rest of the lists are defined below and they are initialized as empty lists.

```
\gdef\@toplist{}
\gdef\@botlist{}
\gdef\@midlist{}
\gdef\@currlist{}
\gdef\@deferlist{}
\gdef\@dbltoplist{}
\gdef\@dbldeferlist{}
```

Example 14.2 Current list

```
\makeatletter
\meaning\@currlist

\meaning \bx@A

\the\bx@A

\the\dimen252

\the\count252

```

```
\the\skip252


\makeatother
```

macro:->
\char"FC
252
0.opt
-1
0.opt

39.15 Page layout parameters

The page layout parameters (all taking values by the standard classes later on are defined here. They are important in building up the page calculations.

```
\newdimen\topmargin
\newdimen\oddsidemargin
\newdimen\evensidemargin
\let\@themargin=\oddsidemargin
\newdimen\headheight
\newdimen\headsep
\newdimen\footskip
\newdimen\textheight
\newdimen\textwidth
\newdimen\columnwidth
\newdimen\columnsep
\newdimen\columnseprule
\newdimen\marginparwidth
\newdimen\marginparsep
\newdimen\marginparpush
```

After these preliminary definitions are made a box is defined to hold material that is inserterted before the dvi file is produced. This is a general hook and widely used by package authors.

\AtBeginDvi{⟨*contents*⟩}

Uses a box register in which to put stuff that must appear before anything else in the **.dvi** file.
The stuff in the box should not add any typeset material to the page when it is unboxed.

```
\newbox\@begindvibox
\def \AtBeginDvi #1{%
  \global \setbox \@begindvibox
    \vbox{\unvbox \@begindvibox #1}%
}
```

\@maxdepth

This is not the right place to set this; it needs to be set in a class/style file when **\maxdepth** is set.
Also, many settings to **\maxdepth** should be to **\@maxdepth**, probably?

```
\newdimen\@maxdepth
\@maxdepth = \maxdepth
```

\paperheight

\paperwidth

Although earlier on, page parameters have been defined, we also need to define the paper height and width.

```
\newdimen\paperheight
\newdimen\paperwidth
```

The following nine switches have to be defined to keep track of various options.

\if@insert

\if@fcolmade

\if@specialpage

\if@firstcolumn

\if@twocolumn

\if@twoside

\if@reversemarginpar

\if@mparswitch

\col@number

Local switches first:

```
\newif \if@insert
```

These should definitely be global:

```
\newif \if@fcolmade
\newif \if@specialpage \@specialpagefalse
```

These should be global but are not always set globally in other files.

```
\newif \if@firstcolumn \@firstcolumntrue
\newif \if@twocolumn   \@twocolumnfalse
```

Not sure about these: two questions. Should things which must apply to a whole doument be local or global (they probably should be 'preamble only' commands)? Are these three such things?

```
\newif \if@twoside      \@twosidefalse
\newif \if@reversemargin \@reversemarginfalse
\newif \if@mparswitch   \@mparswitchfalse
```

This counter has been imported from 'multicol'.

```
\newcount \col@number
\col@number \@ne
```

39.16 Internal registers

```
\newcount\@topnum
\newdimen\@toproom
\newcount\@dbltopnum
\newdimen\@dbltoproom
\newcount\@botnum
\newdimen\@botroom
\newcount\@colnum
\newdimen\@textmin
\newdimen\@fpmin
\newdimen\@colht
\newdimen\@colroom
\newdimen\@pageht
\newdimen\@pagedp
\newdimen\@mparbottom \@mparbottom\z@
\newcount\@currtype
\newbox\@outputbox
\newbox\@leftcolumn
\newbox\@holdpg
```

The page headers and page footers are initialized to their odd values, this makes sense as a document always starts at an odd number.

```
\def\@thehead{\@oddhead} %
\def\@thefoot{\@oddfoot}
```

\clearpage

The tests at the beginning are an experimental attempt to avoid a completely empty page after a `\twocolumn[...]`. This prevents the text from the argument vanishing into a float box, never to be seen again. We hope that it does not produce wrong formatting in other cases.

```
\def\clearpage{%
  \ifvmode
    \ifnum \@dbltopnum =\m@ne
      \ifdim \pagetotal <\topskip
        \hbox{}%
      \fi
    \fi
  \fi
  \newpage
  \write\m@ne{}%
  \vbox{}%
  \penalty -\@Mi
}
```

\cleardoublepage

```
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
    \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
```

\onecolumn

```
\def\onecolumn{%
```

```
  \clearpage
  \global\columnwidth\textwidth
  \global\hsize\columnwidth
  \global\linewidth\columnwidth
  \global\@twocolumnfalse
  \col@number \@ne
  \@floatplacement}
```

\newpage

The two checks at the beginning ensure that an item label or run-in section title immediately before a **\newpage** get printed on the correct page, the one before the page break.

All three tests are largely to make error processing more robust; that is why they all reset the flags explicitly, even when it would appear that this would be done by a **\leavevmode**.

```
\def \newpage {%
  \if@noskipsec
    \ifx \@nodocument\relax
      \leavevmode
      \global \@noskipsecfalse
    \fi
  \fi
  \if@inlabel
    \leavevmode
    \global \@inlabelfalse
  \fi
  \if@nobreak \@nobreakfalse \everypar{}\fi
  \par
  \vfil
  \penalty -\@M}
```

\@emptycol

It may be better to use an invisible rule rather than an empty box here.

```
\def \@emptycol {\vbox{}\penalty -\@M}
```

\twocolumn

\@topnewpage

There are several bug fixes to the two-column stuff here.

```
\def \twocolumn {%
  \clearpage
  \global\columnwidth\textwidth
  \global\advance\columnwidth-\columnsep
  \global\divide\columnwidth\tw@
  \global\hsize\columnwidth
  \global\linewidth\columnwidth
  \global\@twocolumntrue
  \global\@firstcolumntrue
  \col@number \tw@
```

There is no reason to put a **\@dblfloatplacement** here since **\@topnewpage** ignores these settings. The **\@floatplacement** is needed in case this comes after some changes.

```
  \@ifnextchar [\@topnewpage\@floatplacement
}
```

Note that here, getting a box from the freelist can assume success since this comes just after a **\clearpage**.

```
\long\def \@topnewpage [#1]{%
  \@nodocument
  \@next\@currbox\@freelist{}{}%
  \global \setbox\@currbox
    \color@vbox
      \normalcolor
      \vbox {%
        \hsize\textwidth
        \@parboxrestore
        \col@number \@ne
        #1%
        \vskip -\dbltextfloatsep
             }%
    \color@endbox
```

Added size test and warning message; perhaps we should use an error message.

```
  \ifdim \ht\@currbox>\textheight
    \ht\@currbox \textheight
  \fi
```

This next line is not essential but it is more robust to make this value non-zero, in case of weird errors.

This next bit is what is needed from **\@addtodblcol**, plus some extra checks for error trapping.

```
  \global \count\@currbox \tw@
  \@tempdima -\ht\@currbox
  \advance \@tempdima -\dbltextfloatsep
  \global \advance \@colht \@tempdima
  \ifx \@dbltoplist \@empty
  \else
    \@latexerr{Float(s) lost}\@ehb
    \let \@dbltoplist \@empty
  \fi
  \@cons \@dbltoplist \@currbox
```

This setting of **\@dbltopnum** is used only to change the typesetting in **\@combinedblfloats**.

```
  \global \@dbltopnum \m@ne
%<*trace>
    \tr@ce{dbltopnum set to -1 (= \the \@dbltopnum) ↱
          ↳(topnewpage)}%
%</trace>
```

At points such as this we need to check that there is still a minimal amount of room left on the page; this uses an arbitrary small value at present; but note that this value is larger than that used when checking that page is too full of normal floats.

If there is little room left we just force a page-break, OK? This involves producing two empty columns. The second empty column may be produced by `\output`, in which case an extra, misleading, warning will be generated, OK? (This happens only when there is too little room left on the page for any float.) Otherwise (ie if the size is such that it is allowed as a normal float) the extra `\@emptycol` will be invoked in the second column by the conditional code guarded by the `\if@firstcolumn` test.

I now think that the cut-off point here should be `3\baselineskip`, but we make it a bit less so that 3 lines of text will be allowed, OK?

Since this happens only when there is nothing on the page but the 'top-box', the empty box should not cause any problem other than some overfull box messages, which is not entirely misleading.

Here we need two page-ends since both columns need to be empty.

```
  \ifdim \@colht<2.5\baselineskip
    \@latex@warning@no@line {Optional argument of ↩
         ↪\noexpand\twocolumn
                too tall on page \thepage}%
    \@emptycol
    \if@firstcolumn
    \else
      \@emptycol
    \fi
  \else
    \global \vsize \@colht
    \global \@colroom \@colht
    \@floatplacement
  \fi
}
```

39.17 The \output routine

We now arrive at the interesting part. The `\output` is a token register that holds instructions as to how the page is to be typeset. This is called automatically by TEX. Think of it as the main function.

`\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.

The reset of `\par` to the output routine. This avoids problems when the output routine is called within a list where `\par` may be a no-op.

```
\output {%
  \let \par \@@par
  \ifnum \outputpenalty<-\@M
    \@specialoutput
  \else
    \@makecol
    \@opcol
    \@startcolumn
```

```
    \@whilesw \if@fcolmade \fi
      {\@opcol\@startcolumn}%
  \fi
  \ifnum \outputpenalty>-\@Miv
    \ifdim \@colroom<1.5\baselineskip
      \ifdim \@colroom<\textheight
        \@latex@warning@no@line {Text page \thepage\space
                                 contains only floats}%
        \@emptycol
      \else
        \global \vsize \@colroom
      \fi
    \else
      \global \vsize \@colroom
    \fi
  \else
    \global \vsize \maxdimen
  \fi
}
```

\@specialoutput

```
\gdef\@specialoutput{%
   \ifnum \outputpenalty>-\@Mii
     \@doclearpage
   \else
     \ifnum \outputpenalty<-\@Miii
       \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
       \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
     \else
       \global \setbox\@holdpg \vbox{%
                       \unvbox\@holdpg
                       \unvbox\@cclv
                       \setbox\@tempboxa \lastbox
                       \unskip
                                       }%
       \@pagedp \dp\@holdpg
       \@pageht \ht\@holdpg
       \unvbox \@holdpg
       \@next\@currbox\@currlist{%
         \ifnum \count\@currbox>\z@
           \advance \@pageht \@pagedp
           \ifvoid\footins \else
             \advance \@pageht \ht\footins
             \advance \@pageht \skip\footins
             \advance \@pageht \dp\footins
           \fi
           \ifvbox \@kludgeins
             \ifdim \wd\@kludgeins=\z@
               \advance \@pageht \ht\@kludgeins
             \fi
           \fi
```

```
            \@reinserts
            \@addtocurcol
          \else
            \@reinserts
            \@addmarginpar
          \fi
          }\@latexbug
        \ifnum \outputpenalty<\z@
          \if@nobreak
            \nobreak
          \else
            \addpenalty \interlinepenalty
          \fi
        \fi
      \fi
    \fi
}
```

\@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.

```
\def \@doclearpage {%
     \ifvoid\footins
        \ifvbox\@kludgeins
          {\setbox \@tempboxa \box \@kludgeins}%
        \fi
        \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
        \setbox\@tempboxa\box\@cclv
        \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
        \global \let \@toplist \@empty
        \global \let \@botlist \@empty
        \global \@colroom \@colht
        \ifx \@currlist\@empty
        \else
           \@latexerr{Float(s) lost}\@ehb
           \global \let \@currlist \@empty
        \fi
        \@makefcolumn\@deferlist
        \@whilesw\if@fcolmade ↩
              ↪\fi{\@opcol\@makefcolumn\@deferlist}%
        \if@twocolumn
          \if@firstcolumn
            \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
            \global \let \@dbltoplist \@empty
            \global \@colht \textheight
            \begingroup
               \@dblfloatplacement
```

```
                \@makefcolumn\@dbldeferlist
                \@whilesw\if@fcolmade \fi{\@outputpage
                                          \@makefcolumn\@dbldeferlist}%
              \endgroup
            \else
              \vbox{}\clearpage
            \fi
          \fi
        \else
          \setbox\@cclv\vbox{\box\@cclv\vfil}%
          \@makecol\@opcol
          \clearpage
        \fi
}
```

\@opcol

```
\def \@opcol {%
  \if@twocolumn
    \@outputdblcol
  \else
    \@outputpage
  \fi
```

These do not need to be done every time `\@opcol` is used: they should be grouped together since they all need to be done at the end of the non-special output routine, or at the end of a clearpage one.

```
  \global \@mparbottom \z@ \global \@textfloatsheight \z@
  \@floatplacement
}
```

\@makecol

We must rewrite this macro to alllow for variations in page-makeup required by changes in page-length.
This uses a different macro if a special-length column is being produced.

```
\gdef \@makecol {%
   \ifvoid\footins
     \setbox\@outputbox \box\@cclv
   \else
     \setbox\@outputbox \vbox {%
```

This `\boxmaxdepth` setting is to ensure that deep footnotes do not overwrite the footer (on account of the negative skip added later): it should use `\@maxdepth` otherwise the change is pointless when there are footnotes. But see also its use when combining floats.
Macro `\newinsert` computes a number (counting down from 254) and allocates a box, a count, a dimen, and a skip register with that number. The reason for allocating from 254 instead of 255 is that `\box255` is reserved for special OTR use. The reason for allocating downwards is that registers `\count0, \count1`. .. are used for the page number, and many people tend to use registers `\box0, \box1`. .. for temporary storage.

Example 17.3 footins

```
\the\footins

\the\skip\footins

\the\dimen\footins

\the\count\footins
```

253
9.0pt plus 4.0pt minus 2.0pt
578.15999pt
1000

```
      \boxmaxdepth \@maxdepth
      % unbox
      \unvbox \@cclv
      % skip
      \vskip \skip\footins

      \color@begingroup
        \normalcolor
        % draw rule
        \footnoterule

        % unbox footins
        \unvbox \footins
      \color@endgroup
      }%
  \fi
  \let\@elt\relax
  \xdef\@freelist{\@freelist\@midlist}%
  \global \let \@midlist \@empty
  \@combinefloats
  \ifvbox\@kludgeins
    \@makespecialcolbox
  \else
    \setbox\@outputbox \vbox to\@colht {%
      \@texttop
      \dimen@ \dp\@outputbox
      \unvbox \@outputbox
      \vskip -\dimen@
      \@textbottom
      }%
  \fi
  \global \maxdepth \@maxdepth
}
```

\@reinserts

This is the code which reinserts the inserts. It puts them all in one place; this can make some of them come out on the wrong page. It has been put into a separate macro to expedite experimentation.

```
\gdef \@reinserts{%
  \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
}
```

\@makespecialcolbox

This implements certain variations in page-makeup.

```
\gdef \@makespecialcolbox {%
```

First we find the natural height of the column. See above for discussion of what is happening here. This needs further investigation, especially in this 'special case'.

```
   \setbox\@outputbox \vbox {%
     \@texttop
     \dimen@ \dp\@outputbox
     \unvbox\@outputbox
     \vskip-\dimen@
     }%
   \@tempdima \@colht
   \ifdim \wd\@kludgeins>\z@
```

Note that in this case (the *-version), the height of the **\@kludgeins** box is not used since its value is somewhat arbitrary: it need only be big enough to ensure that the page-break is not taken prematurely.

Here we calculate how much vertical space needs to be added in order to enable the column to fit into a box of size **\@colht** using the best information we have about the amount of shrink available (another thing which is known internally about a box, but cannot be accessed at the TEX level!).

This needs TEX3 otherwise **\pageshrink** is zero anyway; it may not be exactly the figure we wish as it is the total available from the all the material collected before the page-break decision is made. It will, we think, always be an overestimate of the actual shrink in the box; therefore this should always force the shortest possible column with the possibility of an overfull box.

This should work for bothe flush- and ragged-bottom setting since it makes the contents no smaller than the size (**\@colht**) of the box into which they are put.

Their should perhaps be an upper limit, of 0pt?, on the extra space added to force shrinking.

See above for a discussion of the **\boxmaxdepth** setting here.

```
     \advance \@tempdima -\ht\@outputbox
     \advance \@tempdima \pageshrink
     \setbox\@outputbox \vbox to \@colht {%
       \unvbox\@outputbox
       \vskip \@tempdima
       \@textbottom
       }%
```

For the unstarred version, the final size of the page is precisely specified. Therefore, at least for the flush-bottom case, we need to ensure that, visually, it has this size exactly.

Thus we calculate this size and set the material in a box of this size, which is then put into a box of size **\@colht** with **\vss** at the bottom.

```
  \else
    \advance \@tempdima -\ht\@kludgeins
```

This type of final packaging could be done always; this may simplify all of this page-makeup.
It is not necessary to set **\boxmaxdepth** here since the **\@outputbox** ends with glue.

```
    \setbox \@outputbox \vbox to \@colht {%
      \vbox to \@tempdima {%
        \unvbox\@outputbox
        \@textbottom}%
      \vss}%
  \fi
```

Finally we need to explicitly make the insert box void.

```
  {\setbox \@tempboxa \box \@kludgeins}%
```

The following macros are just hooks and can be set to add top and bottom glue on all pages.[2][3] They are currently used in the definition of **??**[→ P. ??] and **??**[→ P. ??].

\@texttop

Does nothing by default, otherwise add glue at the top on all pages.

```
 \let \@texttop \relax
```

\@textbottom

Can be used to add glue at the bottom of a page.

```
\let \@textbottom \relax
```

\@resetactivechars

\@activechar@info

added hook to protect against certain active characters in the output routine. Default checks are for active space and end-of-line.

```
\def\@activechar@info #1{%
     \@latex@info@no@line {Active #1 character found while
                           output routine is active
                           \MessageBreak
                           This may be a bug in a package file
                           you are using}%
}
```

Do not put any spaces in this next bit!

[2] http://tex.stackexchange.com/questions/40469/use-of-texttop-and-textbottom-for-vertical-positioning

[3] https://tex.stackexchange.com/questions/131871/vertically-centering-page-with-texttop-and-textbottom

```
\begingroup
\obeylines\obeyspaces%
\catcode`\'\active%
\gdef\@resetactivechars{%
\def^^M{\@activechar@info{EOL}\space}%
\def {\@activechar@info{space}\space}%
\let'\active@math@prime}%
\endgroup
```

\@outputpage

The **\color@hbox** hooks here are used to avoid putting just a colour special into an otherwise empty box (in a header or footer). These boxes are often set to be completely empty and so adding a special produces a very underfull box message.

There has been extensive tidying up of the old code here; including the removal of a level of grouping.

The setting of **\protect** immediately before the **\shipout** is needed so that protected commands within **\write**s are handled correctly.

Within shipout's vbox it is reset to its default value, **\relax**.

Resetting it to its default value after the shipout has been completed (and the contents of the writes have been expanded) must be done by use of **\aftergroup**. This is because it must have the value **\relax** before macros coming from other uses of **\aftergroup** within this box are expanded.

Putting this into the **\aftergroup** token list does not affect the definition used in expanding the **\write**s because the aftergroup token list is only constructed when popping the save-stack, it is not expanded until after the shipout is completed.

Question: should things from an **\aftergroup** within the shipped out box be executed in the environment set up for the writes, or after it finishes?

A lot of this code has been in-lined tp prevent mis-use of internal commands as hooks.

\@outputpage{⟨*void*⟩}

This essentially calls **\shipout**

```
\def\@outputpage{%
\begingroup          % the \endgroup is put in by \aftergroup
```

Now all the set-up stuff has been in-lined for Frank.

First the stuff for the writes.

From here ... was in the command **\@writesetup**.

```
  \let \protect \noexpand
```

RmS 93/08/19: Redefined accents to allow changes in font encoding; but exactly why was this needed?

The **\catcode`\ = 10** was removed as it was considered useless (presumably because nothing gets tokenised during shipout).

This was put in as some error produced active spaces in a mark, I think.

Why was the hyphen reset?

```
  \@resetactivechars
```

If a page break happens between the start of a list and its first item the `@newlist` will be true and this will mess up any list that is used in the header or footer of the page. So we have to reset that flag.

```
\global\let\@@if@newlist\if@newlist
\global\@newlistfalse
```

with the new encoding setup they can use `\let`. It could also use the new internal commands? This next hook replaces the following:

```
\let\-\@dischyph
\let\'\@acci\let\`\@accii\let\=\@acciii
\let\\\@normalcr
\let\par\@@par %% 15 Sep 87 (this was once inside the box)


%       \parindent\z@
%       \parskip\z@skip
%       \everypar{}%
%       \leftskip\z@skip
%       \rightskip\z@skip
%       \parfillskip\@flushglue
%       \lineskip\normallineskip
%       \baselineskip\normalbaselineskip
%       \sloppy
```

```
\@parboxrestore
```

Finally we are ready to `\shipout` the box.

```
\shipout \vbox{%
  \set@typeset@protect
  \aftergroup \endgroup
  \aftergroup \set@typeset@protect
                                % correct? or just restore
                                      by ending
                                % the group?
```

Now the setup inside the shipped out box; this should contain all the stuff that could only affect typesetting; other stuff may need to be reset for the writes also.

From here ... was in the command `\@shipoutsetup`.

The `\@specialpage` picks up the pagestyle using the `ps@`

```
\if@specialpage
  \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
\fi
\if@twoside
  \ifodd\count\z@ \let\@thehead\@oddhead
        \let\@thefoot\@oddfoot
        \let\@themargin\oddsidemargin
  \else \let\@thehead\@evenhead
     \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
  \fi
\fi
```

The rest was always inside the box.

```
\reset@font
```

RmS 93/08/06 Added `\lineskiplimit=0pt` to guard against it being non-zero: e.g. by `\offinterlineskip` being in effect.

There are probably lots of other things that may need resetting.

```
\normalsize
```

Reset the space factors. Call `\normalsfcodes` (from patch file) latex/2404

```
\normalsfcodes
```

Reset these here (previously reset separately for head and foot)

```
\let\label\@gobble
\let\index\@gobble
\let\glossary\@gobble
```

```
\baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
```

... to here was in the command `\@shipoutsetup`.

```
  \@begindvi ❶
  \vskip \topmargin
  \moveright\@themargin \vbox {%
    \setbox\@tempboxa \vbox to\headheight{%
      \vfil
      \color@hbox
        \normalcolor
        \hb@xt@\textwidth{\@thehead}%
      \color@endbox
      }%                          %% 22 Feb 87
    \dp\@tempboxa \z@
    \box\@tempboxa
    \vskip \headsep
    \box\@outputbox
    \baselineskip \footskip
    \color@hbox
      \normalcolor
      \hb@xt@\textwidth{\@thefoot}%
    \color@endbox
    }%
  }%
```

`\endgroup` now inserted by `\aftergroup`

Restore `\if@newlist`

```
\global\let\if@newlist\@@if@newlist
```

```
\global \@colht \textheight
\stepcounter{page}%
```

It is now clear that this does something useful, thanks to Piet van Oostrum. It is needed because a float page is made without using TeX's page-builder; thus the output routine is never called so the marks are not updated.

```
\let\firstmark\botmark
}
```

At ❶ the `@begindvi` was inserted for the material taht it holds. This in reality is a hook, used to place anything we want at the page, provided it holds material that do not take any space.

53.0pt

HANG 39. ltoutput.dtx 487

February 1, 2019 phddoc

\@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. Provided that you inserting material which is absolutely positioned, this is a good place to hook-in.

```
\def \@begindvi{%
  \unvbox \@begindvibox
  \global\let \@begindvi \@empty
}
```

The combining of the floats happens here.

\@combinefloats

\@cflb

The **\boxmaxdepth** setting here was not made local to a box so was dangerous. It is needed only within the box made by **\@cflt** (and not normally even there), so it has been moved there; this also agrees with the original pseudcode.
The **\@combinefloat** function will add the top and bottom floats if their lists are not empty. **\cf**⟨*lt*⟩ is or top and **lb** for bottom.

```
\def \@combinefloats {%
    \ifx \@toplist\@empty \else \@cflt \fi
    \ifx \@botlist\@empty \else \@cflb \fi
}
```

First define the **\@elt**

```
\def \@cflt{%
    \let \@elt \@comflelt
    \setbox\@tempboxa \vbox{}%
    \@toplist
    \setbox\@outputbox \vbox{%
                              \boxmaxdepth \maxdepth
                              \unvbox\@tempboxa
                              \vskip -\floatsep
                              \topfigrule
                              \vskip \textfloatsep
                              \unvbox\@outputbox
                              }%

    % reset |\@elt| to |\relax|
    \let\@elt\relax

    % return |\@toplist| to the |\@freelist|
    \xdef\@freelist{\@freelist\@toplist}%
```

```

    % empty the |\@toplist|
    \global\let\@toplist\@empty
}
\def \@cflb {%
    \let\@elt\@comflelt
    \setbox\@tempboxa \vbox{}%
    \@botlist
    \setbox\@outputbox \vbox{%
                              \unvbox\@outputbox
                              \vskip \textfloatsep
                              \botfigrule
                              \unvbox\@tempboxa
                              \vskip -\floatsep
                              }%
    \let\@elt\relax
    \xdef\@freelist{\@freelist\@botlist}%
    \global \let \@botlist\@empty
}
```

\@comflelt

\@comdblflelt

\@combinedblfloats

```
\def\@comflelt#1{\setbox\@tempboxa
      \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
\def\@comdblflelt#1{\setbox\@tempboxa
      \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
\def \@combinedblfloats{%
  \ifx \@dbltoplist \@empty
  \else
    \setbox\@tempboxa \vbox{}%
    \let \@elt \@comdblflelt
    \@dbltoplist
    \let \@elt \relax
    \xdef \@freelist {\@freelist\@dbltoplist}%
    \global\let \@dbltoplist \@empty
    \setbox\@outputbox \vbox to\textheight
      {%\boxmaxdepth\maxdepth   %% probably not needed, CAR
       \unvbox\@tempboxa\vskip-\dblfloatsep
       \ifnum \@dbltopnum>\m@ne
         \dblfigrule
       \fi
       \vskip \dbltextfloatsep
       \box\@outputbox
       }%
  \fi
}
```

\@startcolumn

We could combine (most of) these two into \@startcol <list>. This is not quite as efficient but it now has the same structure as \@startdblcolumn. The empty-list test has been moved to \@tryfcolumn.

```
\def \@startcolumn {%
  \global \@colroom \@colht
  \@tryfcolumn \@deferlist
  \if@fcolmade
%<*trace>
    \tr@ce{PAGE: float \if@twocolumn column \else page \fi
                 completed}%
%</trace>
  \else
    \begingroup
      \let \reserved@b \@deferlist
      \global \let \@deferlist \@empty
      \let \@elt \@scolelt
      \reserved@b
    \endgroup
  \fi
}
\def \@startdblcolumn {%
% \global \@colht \textheight
  \@tryfcolumn \@dbldeferlist
  \if@fcolmade
%<*trace>
    \tr@ce{PAGE: double float page completed}%
%</trace>
  \else
    \begingroup
      \let \reserved@b \@dbldeferlist
      \global \let \@dbldeferlist \@empty
      \let \@elt \@sdblcolelt
      \reserved@b
    \endgroup
  \fi
}
```

\@tryfcolumn

Now tests if its list is empty before any further exertion.

```
\def \@tryfcolumn #1{%
  \global \@fcolmadefalse
  \ifx #1\@empty
  \else
%<*trace>
     \tr@ce{PAGE: try float \if@twocolumn column/page\else ↊
          ↋page\fi
                  ---\string #1}%
     \tr@ce{----- \string #1: #1}%
%</trace>
```

```
    \xdef\@trylist{#1}%
    \global \let \@failedlist \@empty
    \begingroup
      \let \@elt \@xtryfc \@trylist
    \endgroup
    \if@fcolmade
      \@vtryfc #1%
    \fi
  \fi
}
```

\@scolelt

```
\def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}
```

\@sdblcolelt

```
\def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}
```

\@vtryfc

```
\def\@vtryfc #1{%
  \global\setbox\@outputbox\vbox{}%
  \let\@elt\@wtryfc
  \@flsucceed
  \global\setbox\@outputbox \vbox to\@colht{%
    \vskip \@fptop
    \vskip -\@fpsep
    \unvbox \@outputbox
    \vskip \@fpbot}%
  \let\@elt\relax
  \xdef #1{\@failedlist\@flfail}%
  \xdef\@freelist{\@freelist\@flsucceed}}
```

\@wtryfc

```
\def\@wtryfc #1{%
  \global\setbox\@outputbox\vbox{%
    \unvbox\@outputbox
    \vskip\@fpsep
    \box #1}}
```

\@xtryfc

```
\def\@xtryfc #1{%
  \@next\reserved@a\@trylist{}{}%
  \@currtype \count #1%
  \divide\@currtype\@xxxii
  \multiply\@currtype\@xxxii
  \@bitor \@currtype \@failedlist
  \@testfp #1%
  \ifdim \ht #1>\@colht
    \@testtrue
  \fi
```

```
  \if@test
    \@cons\@failedlist #1%
  \else
    \@ytryfc #1%
  \fi}
```

\@ytryfc

```
\def\@ytryfc #1{%
  \begingroup
    \gdef\@flsucceed{\@elt #1}%
    \global\let\@flfail\@empty
    \@tempdima\ht #1%
    \let\@elt\@ztryfc
    \@trylist
    \ifdim \@tempdima >\@fpmin
      \global\@fcolmadetrue
    \else
      \@cons\@failedlist #1%
    \fi
  \endgroup
  \if@fcolmade
    \let\@elt\@gobble
  \fi}
```

\@ztryfc

```
\def\@ztryfc #1{%
  \@tempcnta \count#1%
  \divide\@tempcnta\@xxxii
  \multiply\@tempcnta\@xxxii
  \@bitor \@tempcnta {\@failedlist \@flfail}%
  \@testfp #1%
  \@tempdimb\@tempdima
  \advance\@tempdimb \ht#1%
  \advance\@tempdimb\@fpsep
  \ifdim \@tempdimb >\@colht
    \@testtrue
  \fi
  \if@test
    \@cons\@flfail #1%
  \else
    \@cons\@flsucceed #1%
    \@tempdima\@tempdimb
  \fi}
```

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

```
%<*2ekernel|autoload|fltrace>
\def \@addtobot {%
   \@getfpsbit 4\relax
```

```
   \ifodd \@tempcnta
     \@flsetnum \@botnum
     \ifnum \@botnum>\z@
       \@tempswafalse
       \@flcheckspace \@botroom \@botlist
       \if@tempswa
         \global \maxdepth \z@
         \@flupdates \@botnum \@botroom \@botlist
         \@inserttrue
       \fi
%<*trace>
     \else
       \tr@ce{Fail: botnum = \the \@botnum:
                                  fpstype \the \@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \tr@ce{ERROR: !b float not successful (addtobot)}%
       \fi
%</trace>
     \fi
   \fi
}
```

\@addtotoporbot

```
\def \@addtotoporbot {%
%<*trace>
   \tr@ce{***Start addtotoporbot}%
%</trace>
   \@getfpsbit \tw@
%<*trace>
   \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi top:
                                                      \the ↩
                                                           ↪\@fpstype}%
%</trace>
   \ifodd \@tempcnta
     \@flsetnum \@topnum
     \ifnum \@topnum>\z@
       \@tempswafalse
       \@flcheckspace \@toproom \@toplist
       \if@tempswa
         \@bitor\@currtype{\@midlist\@botlist}%
%<*trace>
           \tr@ce{(mid+bot)list: \@midlist, \@botlist:
                                 (addtotoporbot-before)}%
%</trace>
         \if@test
%<*trace>
           \tr@ce{type already on list: mid or bot---sent to ↩
                 ↪addtobot}%
%</trace>
         \else
          \@flupdates \@topnum \@toproom \@toplist
%<*trace>
          \tr@ce{colroom (after-top) = \the \@colroom}%
```

```
          \tr@ce{colnum (after-top) = \the \@colnum}%
          \tr@ce{topnum (after-top) = \the \@topnum}%
          \tr@ce{***Success: top}%
%</trace>
          \@inserttrue
         \fi
       \fi
%<*trace>
     \else
       \tr@ce{Fail: topnum = \the \@topnum: fpstype
                                                 \the ⤸
                                                      ↳\@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \tr@ce{ERROR: !t float not successful ⤸
                 ↳(addtotoporbot)}%
       \fi
%</trace>
     \fi
   \fi
   \if@insert
   \else
%<*trace>
     \tr@ce{sent to addtobot (addtotoporbot)}%
%</trace>
     \@addtobot
   \fi
}
%</2ekernel|autoload|fltrace>
```

```
%<*2ekernel|autoload|fltrace|flafter>
\def \@addtocurcol {%
%<*trace>
  \tr@ce{***Start addtocurcol}%
%</trace>
   \@insertfalse
   \@setfloattypecounts
   \ifnum \@fpstype=8
%<*trace>
     \tr@ce{fpstype !p only (addtocurcol): \the \@fpstype = ⤸
            ↳8?}%
%</trace>
   \else
     \ifnum \@fpstype=24
%<*trace>
       \tr@ce{fpstype p only (addtocurcol): \the \@fpstype = ⤸
              ↳24?}%
%</trace>
     \else
       \@flsettextmin
%<*trace>
       \tr@ce{textfloatsheight (before) = \the ⤸
              ↳\@textfloatsheight}%
%</trace>
       \advance \@textmin \@textfloatsheight
```

```
        \@reqcolroom \@pageht
%        \advance \@reqcolroom \@pagedp
%<*trace>
        \tr@ce{textmin + textfloatsheight: \the \@textmin}%
        \tr@ce{page-so-far: \the \@reqcolroom}%
%</trace>
        \ifdim \@textmin>\@reqcolroom
          \@reqcolroom \@textmin
%<*trace>
          \tr@ce{ORD? textmin being used}%
%</trace>
        \fi
        \advance \@reqcolroom \ht\@currbox
%<*trace>
        \tr@ce{float size = \the \ht \@currbox (addtocurcol)}%
        \tr@ce{colroom = \the \@colroom (addtocurcol)}%
        \tr@ce{reqcolroom = \the \@reqcolroom (addtocurcol)}%
%</trace>
        \ifdim \@colroom>\@reqcolroom
          \@flsetnum \@colnum
          \ifnum \@colnum>\z@
            \@bitor\@currtype\@deferlist
%<*trace>
            \tr@ce{deferlist: \@deferlist: ↱
                 ↳(addtocurcol-before)}%
%</trace>
            \if@test
%<*trace>
              \tr@ce{type already on list: defer (addtocurcol)}%
%</trace>
            \else
              \@bitor\@currtype\@botlist
%<*trace>
            \tr@ce{botlist: \@botlist: (addtocurcol-before)}%
%</trace>
              \if@test
%<*trace>
                \tr@ce{type already on list: bot---sent to ↱
                     ↳addtobot}%
%</trace>
                \@addtobot
              \else
%<*trace>
                \tr@ce{fpstype \ifodd \@tempcnta OK \else not ↱
                     ↳\fi
                       here: \the \@fpstype}%
%</trace>
                \ifodd \count\@currbox
                  \advance \@reqcolroom \intextsep
                  \ifdim \@colroom>\@reqcolroom
                    \global \advance \@colnum \m@ne
                    \global \advance \@textfloatsheight ↱
                         ↳\ht\@currbox
```

```
                    \global \advance \@textfloatsheight ↩
                         ↪2\intextsep
                    \@cons \@midlist \@currbox
%<*trace>
                    \tr@ce{***Success: here}%
                    \tr@ce{textfloatsheight (after-here) =
                         \the \@textfloatsheight}%
                    \tr@ce{colnum (after-here) = \the \@colnum}%
%</trace>
                    \if@nobreak
                      \nobreak
                      \@nobreakfalse
                      \everypar{}%
                    \else
                      \addpenalty \interlinepenalty
                    \fi
                    \vskip \intextsep
                    \box\@currbox
                    \penalty\interlinepenalty
                    \vskip\intextsep
                    \ifnum\outputpenalty <-\@Mii \vskip ↩
                         ↪-\parskip\fi
                    \outputpenalty \z@
                    \@insertrue
%<*trace>
                  \else
                    \tr@ce{Fail---no room at 2nd test of colroom
                                 (addtocorcol ↩
                                      ↪\string\intextsep)}%
%</trace>
                  \fi
                \fi
                \if@insert
                \else
%<*2ekernel|autoload|fltrace>
%<*trace>
                  \tr@ce{not here: sent to addtotoporbot}%
%</trace>
                  \@addtotoporbot
%</2ekernel|autoload|fltrace>
%<*!2ekernel&!autoload&!fltrace>
%<*trace>
                  \tr@ce{not here: sent to addtobot}%
%</trace>
                  \@addtobot
%</!2ekernel&!autoload&!fltrace>
                \fi
              \fi
            \fi
%<*trace>
          \else
            \tr@ce{Fail: colnum = \the \@colnum:
                         fpstype \the \@fpstype=ORD?}%
```

```
            \ifnum \@fpstype<\sixt@@n
              \tr@ce{ERROR: BANG float not successful (addtocurcol)}%
            \fi
%</trace>
          \fi
%<*trace>
        \else
          \tr@ce{Fail---no room: fl box ht: \the \ht \@currbox
                                                    (addtocurcol)}%
%</trace>
        \fi
      \fi
    \fi
    \if@insert
    \else
      \@resethfps
%<*trace>
      \tr@ce{put on deferlist (addtocurcol)}%
%</trace>
      \@cons\@deferlist\@currbox
%<*trace>
      \tr@ce{deferlist: \@deferlist: (addtocurcol-after)}%
%</trace>
    \fi
}
%</2ekernel|autoload|fltrace|flafter>
```

\@addtonextcol

```
%<*2ekernel|autoload|fltrace>
\def\@addtonextcol{%
  \begingroup
%<*trace>
    \tr@ce{***Start addtonextcol}%
%</trace>
    \@insertfalse
    \@setfloattypecounts
    \ifnum \@fpstype=8
%<*trace>
      \tr@ce{fpstype not curcol: \the \@fpstype = 8?}%
%</trace>
    \else
      \ifnum \@fpstype=24
%<*trace>
        \tr@ce{fpstype not curcol: \the \@fpstype = 24?}%
%</trace>
      \else
        \@flsettextmin
%<*trace>
        \tr@ce{text-so-far: 0pt (top of col)}%
%</trace>
        \@reqcolroom \ht\@currbox
%<*trace>
```

```
        \tr@ce{float size: \the \@reqcolroom (addtonextcol)}%
%</trace>
        \advance \@reqcolroom \@textmin
%<*trace>
        \tr@ce{colroom = \the \@colroom (addtonextcol)}%
        \tr@ce{reqcolroom = \the \@reqcolroom (addtonextcol)}%
%</trace>
        \ifdim \@colroom>\@reqcolroom
          \@flsetnum \@colnum
          \ifnum\@colnum>\z@
             \@bitor\@currtype\@deferlist
%<*trace>
             \tr@ce{deferlist: \@deferlist: ↱
                  ↳(addtonextcol-before)}%
%</trace>
             \if@test
%<*trace>
               \tr@ce{type already on list: defer ↱
                    ↳(addtonextcol)}%
%</trace>
             \else
%<*trace>
               \tr@ce{sent to addtotoporbot (addtonextcol)}%
%</trace>
               \@addtotoporbot
             \fi
          \fi
%<*trace>
        \else
          \tr@ce{Fail---no room: fl box ht: \the \ht \@currbox
                                                  (addtonextcol)}%
%</trace>
        \fi
      \fi
    \fi
    \if@insert
    \else
%<*trace>
      \tr@ce{put back on deferlist (addtonextcol)}%
%</trace>
      \@cons\@deferlist\@currbox
%<*trace>
      \tr@ce{deferlist: \@deferlist: (addtonextcol-after)}%
%</trace>
    \fi
%<*trace>
    \tr@ce{End of addtonextcol -- locally counts:}%
    \tr@ce{ col: \the \@colnum. top: \the \@topnum. bot: \the ↱
          ↳\@botnum.}%
%</trace>
  \endgroup
%<*trace>
  \tr@ce{End of addtonextcol -- globally counts:}%
  \tr@ce{col: \the \@colnum. top: \the \@topnum. bot: \the ↱
```

```
          ↊\@botnum.}%
%</trace>
}
\def\@addtodblcol{%
  \begingroup
%<*trace>
  \tr@ce{***Start addtodblcol}%
%</trace>
   \@insertfalse
   \@setfloattypecounts
   \@getfpsbit \tw@
%<*trace>
   \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi dbltop:
                                                        \the ↊
                                                              ↊\@fpstype}%
%</trace>
   \ifodd\@tempcnta
     \@flsetnum \@dbltopnum
     \ifnum \@dbltopnum>\z@
       \@tempswafalse
       \ifdim \@dbltoproom>\ht\@currbox
         \@tempswatrue
%<*trace>
         \tr@ce{Space OK: \@dbltoproom =
                \the \@dbltoproom > \the \ht \@currbox
                                        (dbltoproom)}%
%</trace>
       \else
%<*trace>
         \tr@ce{fpstype: \the \@fpstype (addtodblcol)}%
%</trace>
         \ifnum \@fpstype<\sixt@@n
%<*trace>
           \tr@ce{BANG float ignoring \@dbltoproom}%
           \tr@ce{\@spaces \@dbltoproom = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox-BANG}%
%</trace>
           \advance \@dbltoproom \@textmin
%<*trace>
           \tr@ce{Local value of texmin: \the\@textmin}%
           \tr@ce{\@spaces space on page = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox-BANG}%
%</trace>
           \ifdim \@dbltoproom>\ht\@currbox
             \@tempswatrue
%<*trace>
             \tr@ce{Space OK BANG: space on page = \the ↊
                  ↊\@dbltoproom >
                                                 \the \ht ↊
                                                       ↊\@currbox}%
           \else
             \tr@ce{fpstype: \the \@fpstype}%
             \tr@ce{Fail---no room dbltoproom-BANG?:}%
```

```
              \tr@ce{\@spaces space on page = \the \@dbltoproom.
                             Ht float: \the \ht \@currbox}%
%</trace>
            \fi
            \advance \@dbltoproom -\@textmin
%<*trace>
          \else
            \tr@ce{fpstype: \the \@fpstype}%
            \tr@ce{Fail---no room dbltoproom-ORD?:}%
            \tr@ce{\@spaces \@dbltoproom = \the \@dbltoproom.
                             Ht float: \the \ht \@currbox}%
%</trace>
          \fi
        \fi
        \if@tempswa
            \@bitor \@currtype \@dbldeferlist
%<*trace>
            \tr@ce{dbldeferlist: \@dbldeferlist: (before)}%
%</trace>
            \if@test
%<*trace>
               \tr@ce{type already on list: dbldefer}%
%</trace>
            \else
               \@tempdima -\ht\@currbox
               \advance\@tempdima
                 -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
                                                \dblfloatsep \fi
               \global \advance \@dbltoproom \@tempdima
               \global \advance \@colht \@tempdima
               \global \advance \@dbltopnum \m@ne
               \@cons \@dbltoplist \@currbox
%<*trace>
               \tr@ce{dbltopnum (after) = \the \@dbltopnum}%
               \tr@ce{***Success: dbltop}%
%</trace>
               \@inserttrue
            \fi
        \fi
%<*trace>
      \else
        \tr@ce{Fail: dbltopnum = \the \@dbltopnum: fpstype
                                                  \the \@fpstype=ORD?}%
        \ifnum \@fpstype<\sixt@@n
          \tr@ce{ERROR: !t float not successful (addtodblcol)}%
        \fi
%</trace>
      \fi
    \fi
    \if@insert
    \else
%<*trace>
```

```
      \tr@ce{put on dbldeferlist}%
%</trace>
      \@cons\@dbldeferlist\@currbox
%<*trace>
      \tr@ce{dbldeferlist: \@dbldeferlist: (after)}%
%</trace>
    \fi
%<*trace>
    \tr@ce{End of addtodblcol -- locally count:}%
    \tr@ce{ dbltop: \the \@dbltopnum.}%
%</trace>
  \endgroup
%<*trace>
  \tr@ce{End of addtodblcol -- globally count:}%
  \tr@ce{dbltop: \the \@dbltopnum.}%
%</trace>
}
%</2ekernel|autoload|fltrace>
```

\@addmarginpar

Defining the marginpar placement function

```
%<*2ekernel|autoload>
\def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
    \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
    \if@twocolumn
        \if@firstcolumn \@tempcnta\m@ne \fi
    \else
      \if@mparswitch
         \ifodd\c@page \else\@tempcnta\m@ne \fi
      \fi
      \if@reversemargin \@tempcnta -\@tempcnta \fi
    \fi
    \ifnum\@tempcnta <\z@ ⤸
         ↳\global\setbox\@marbox\box\@currbox \fi
    \@tempdima\@mparbottom
    \advance\@tempdima -\@pageht
    \advance\@tempdima\ht\@marbox
    \ifdim\@tempdima >\z@
      \@latex@warning@no@line {Marginpar on page ⤸
            ↳\thepage\space moved}%
    \else
      \@tempdima\z@
    \fi
    \global\@mparbottom\@pageht
    \global\advance\@mparbottom\@tempdima
    \global\advance\@mparbottom\dp\@marbox
    \global\advance\@mparbottom\marginparpush
    \advance\@tempdima -\ht\@marbox
```

Putting box movement inside the ‘marbox’:

```
    \global\setbox \@marbox
                   \vbox {\vskip \@tempdima
                          \box \@marbox}%
```

```
    \global \ht\@marbox \z@
    \global \dp\@marbox \z@
    \kern -\@pagedp
    \nointerlineskip
    \hb@xt@\columnwidth
      {\ifnum \@tempcnta >\z@
          \hskip\columnwidth \hskip\marginparsep
       \else
          \hskip -\marginparsep \hskip -\marginparwidth
       \fi
       \box\@marbox \hss}%
    \nointerlineskip
    \hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}
%</2ekernel|autoload>
```

39.17.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for LaTeX2e.

`\enlargethispage{<dim>}`

Adds `<dim>` to the height of the current column only. On the printed page the bottom of this column is extended downwards by exactly `<dim>` without having any effect on the placement of the footer; this may result in an overprinting.

`\enlargethispage*{<dim>}`

Similar to `\enlargethispage` but it tries to squeeze the column to be printed in as small a space as possible, ie it uses any shrinkability in the column. If the column was not explicitly broken (*e.g.* with `\pagebreak`) this may result in an overfull box message but execpt for this it will come out as expected (if you know what to expect).

The star form of this command is dedicated to Leslie Lamport, the other we need for ourselves (FMi, CAR).

These commands may well have unwanted effects if used soon before a `\clearpage`: please give keep them clear of such places.

`\@kludgeins`

The insert which makes TeX do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```
%<*2ekernel|def1>
\newinsert \@kludgeins
\global\dimen\@kludgeins \maxdimen
\global\count\@kludgeins 1000
%</2ekernel|def1>
```

```
%<*2ekernel|def1>
\gdef \enlargethispage {%
   \@ifstar
     {%
%<*trace>
      \tr@ce{Enlarging page height * }%
%</trace>
      \@enlargepage{\hbox{\kern\p@}}}%
     {%
%<*trace>
      \tr@ce{Enlarging page height exactly---}%
%</trace>
      \@enlargepage\@empty}%
}
%</2ekernel|def1>
%<*autoload>
\def\enlargethispage{\@autoload{out1}\enlargethispage}
%</autoload>

%<*2ekernel|def1>
\gdef\@enlargepage#1#2{%
%<*trace>
   \tr@ce{\@spaces\@spaces by #2}%
%</trace>
   \@tempskipa#2\relax
   \ifdim \@tempskipa>.5\maxdimen
     \@latexerr{Suggested\space extra\space height\space
                (\the\@tempskipa)\space dangerously\space
                large}\@eha
   \else
     \ifdim \vsize<.5\maxdimen
%<*trace>
       \tr@ce {Kludgeins added--pagegoal before: \the\pagegoal}%
%</trace>
       \@bsphack
         \insert\@kludgeins{#1\vskip-\@tempskipa}%
       \@esphack

%<*trace>
       \ifvmode \par
         \tr@ce {Kludgeins added--pagegoal after: \the\pagegoal}%
       \fi
%</trace>
     \else
       \@latexerr{Page\space height\space already\space
                  too\space large}\@eha
     \fi
   \fi
}
%</2ekernel|def1>
```

39.18 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.

```
%<*trace>
\def \@tracemessage #1{\typeout{LaTeX2e: #1}}
\def \tracefloats{\let \tr@ce \@tracemessage}
\def \notrace {\let \tr@ce \@gobble}
\notrace
\def \@traceval #1{\tr@ce{\string #1 = \the #1}}
\def \tracefloatvals{%
  \@dblfloatplacement
  \@floatplacement
  \@traceval\@colnum
  \@traceval\@colroom
  \@traceval\@topnum
  \@traceval\@toproom
  \@traceval\@botnum
  \@traceval\@botroom
  \@traceval\@fpmin
  \tr@ce{\string\textfraction = \textfraction}%
  \@traceval\@dbltopnum
  \@traceval\@dbltoproom
}
%</trace>
%<*flafter>
\providecommand\tr@ce[1]{}
%</flafter>
```

`\suppressfloats`

`\@flstop`

Float suppression commands: these set the relevant counter globally to zero. Thus they are overridden for a particular float by an ! specifier.

```
\def \suppressfloats {%
   \@ifnextchar [%
     \@flstop
    {\global \@colnum \z@}%
}
```

Maybe this should be a loop over `#1`?

```
\def \@flstop [#1]{%
   \if t#1%
     \global \@topnum \z@
   \fi
   \if b#1%
     \global \@botnum \z@
   \fi
}
```

Manipulation of float placement and type; both their strings and the corresponding count registers.

\@fpstype

\@reqcolroom

\@textfloatsheight

First a new count register to go with **\@currtype**.
Then a new skip register, for information needed to remove the **\@maxsep** conservatism: it is possible that this could use a temporary register.
Finally a dimension register to hold the total height of in-text floats on the current page. This is needed to implement a major change in the functionality of **\@addtocurcol** which is, nevertheless, a bug fix. It is not local and therefore cannot be a temporary register.

```
\newcount \@fpstype
\newdimen \@reqcolroom
\newdimen \@textfloatsheight
```

```
%<*2ekernel|autoload|fltrace>
\def \@fpsadddefault {%
%<*trace>
   \tr@ce{fps changed from: \@fps}%
%</trace>
   \edef \@fps {\@fps\csname fps@\@captype \endcsname}%
   \@latex@warning {%
     No positions in optional float specifier.\MessageBreak
     Default added (so using `\@fps')}%
}
```

```
\def \@setfloattypecounts {%
  \@currtype \count\@currbox
  \@fpstype \count\@currbox
  \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
  \advance \@fpstype -\@currtype
%<*trace>
  \tr@ce{(mod 32) fpstype: \the \@fpstype}%
  \tr@ce{(mult of 32) currtype: \the \@currtype}%
% Tracing only: but some should be changed into real errors/warnings?
  \ifnum \@fpstype<\sixt@@n
    \ifnum \@fpstype=\z@
      \tr@ce{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
    \fi
    \ifnum \@fpstype=\@ne
      \tr@ce{WARNING: only h, fpstype = \the \@fpstype = 1?}%
    \fi
    \tr@ce{BANG float}%
  \else
    \ifnum \@fpstype=\sixt@@n
```

```
      \tr@ce{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
    \fi
    \ifnum \@fpstype=17
      \tr@ce{WARNING: only h, fpstype = \the \@fpstype = 17?}%
    \fi
    \tr@ce{ORD float}%
  \fi
%</trace>
}
```

Macros for getting, testing and setting bits of the fps.

\@getfpsbit

Sets **\@tempcnta** to required bit of **\count\@currbox**.

```
\def \@getfpsbit {%
   \@boxfpsbit \@currbox
}
```

\@boxfpsbit

Used above.

```
\def \@boxfpsbit #1#2{%
   \@tempcnta \count#1%
   \divide \@tempcnta #2\relax
}
```

\@testfp

New definition of the float page test.

```
\def \@testfp #1{%
   \@boxfpsbit #18\relax % Really `#1 8' for human readers!
   \ifodd \@tempcnta
   \else
     \@testtrue
   \fi
}
```

\@setfpsbit

Sets required bit of **\@tempcnta** (to 1). [4]. This is used earlier by the @float command to set the bitset of a particular float.

```
\def \@setfpsbit #1{%
   \@tempcntb \@tempcnta
   \divide \@tempcntb #1\relax
   \ifodd \@tempcntb
   \else
     \advance \@tempcnta #1\relax
   \fi
}
```

[4] v1.0f,1993/12/05,Command added.

\@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**.

```
\def \@resethfps {%
   \let\reserved@a\@empty
   \ifnum \@fpstype=\@ne
      \def \reserved@a {!}%
      \@fpstype 17
   \fi
   \ifnum \@fpstype=17
     \global \advance \count\@currbox \tw@
     \@latex@warning@no@line {%
       `\reserved@a h' float specifier changed to ↩
            ↪`\reserved@a ht'}%
   \fi
}
```

Special stuff for BANG floats.

\@flsetnum

Ignores any zero float counter value in case BANG.
It uses a local assignment to the normally global counter: a bit naughty, perhaps?
These assgnments are safe so long as the counter involved is only consulted once (*i.e* only for the 'bang float') with the changed value. This is the case within **\@addtocurcol** because it is used only once within a call of the output routine (which forms a group).
For **\@addtonextcol** this is achieved by putting a group around its code; this is needed because it is called (by **\@startcolumn**) for each float which was on the deferlist. Almost identical considerations pertain to **\@addtodblcol**. There may be more efficient ways to handle this, but the group seems to be the simplest.

```
\def \@flsetnum #1{%
%<*trace>
   \tr@ce{fpstype: \the \@fpstype (flsetnum \string#1)}%
%</trace>
   \ifnum \@fpstype<\sixt@@n
     \ifnum #1=\z@
%<*trace>
       \tr@ce{BANG float resetting \string#1 to 1}%
%</trace>
       #1\@ne
     \fi
   \fi
%<*trace>
   \tr@ce{#1 (before) = \the #1}%
%</trace>
}
```

\@flsettextmin

This ignores **\textfraction** space restriction in case BANG.

```
\def \@flsettextmin {%
%<*trace>
   \tr@ce{fpstype: \the \@fpstype (flsettextmin)}%
%</trace>
   \ifnum \@fpstype<\sixt@@n
%<*trace>
     \tr@ce{BANG ignoring textmin}%
%</trace>
     \@textmin \z@
   \else
     \@textmin \textfraction\@colht
%<*trace>
     \tr@ce{ORD textmin = \the \@textmin}%
%</trace>
   \fi
}
```

\@flcheckspace

This ignores space restriction in case BANG; this is still slightly conervative since it does not allow for the fact that, if there is no text in the column then **\textfloatsep** is not needed. Sets **@tempswa** true if there is room for **\@currbox**.

```
\def \@flcheckspace #1#2{%
   \advance \@reqcolroom
     \ifx #2\@empty \textfloatsep \else \floatsep \fi
%<*trace>
   \tr@ce{colroom = \the \@colroom (flcheckspace \string#1 ↊
        ↊\string#2)}%
   \tr@ce{reqcolroom = \the \@reqcolroom
                                    (flcheckspace \string#1 ↊
                                         ↊\string#2)}%
%</trace>
   \ifdim \@colroom>\@reqcolroom
     \ifdim #1>\ht\@currbox
       \@tempswatrue
%<*trace>
       \tr@ce{Space OK: #1 = \the #1 > \the \ht \@currbox
                                    (flcheckspace \string#1 ↊
                                         ↊\string#2)}%
%</trace>
     \else
%<*trace>
       \tr@ce{fpstype: \the \@fpstype
                                    (flcheckspace \string#1 ↊
                                         ↊\string#2)}%
%</trace>
       \ifnum \@fpstype<\sixt@@n
%<*trace>
         \tr@ce{BANG float ignoring #1
```

```
                                    (flcheckspace \string#1 ↊
                                         ↊\string#2):}%
          \tr@ce{\@spaces #1 = \the #1.  Ht float: \the \ht ↊
                ↊\@currbox
                                                          BANG}%
%</trace>
          \@tempswatrue
%<*trace>
        \else
          \tr@ce{Fail---no room (flcheckspace \string#1 ↊
                ↊\string#2)
                        (fpstype \the \@fpstype=ORD?):}%
          \tr@ce{\@spaces #1 = \the #1.  Ht float: \the \ht ↊
                ↊\@currbox
                                                          ORD?}%
%</trace>
        \fi
      \fi
%<*trace>
    \else
      \tr@ce{Fail---no room at 2nd test of colroom
                     (flcheckspace \string#1 \string#2)}%
%</trace>
    \fi
}
```

\@flupdates

This updates everything when a float is placed.

```
\def \@flupdates #1#2#3{%
   \global \advance #1\m@ne
   \global \advance \@colnum \m@ne
   \@tempdima -\ht\@currbox
   \advance \@tempdima
     -\ifx #3\@empty \textfloatsep \else \floatsep \fi
   \global \advance #2\@tempdima
   \global \advance \@colroom \@tempdima
   \@cons #3\@currbox
}
```

Interesting facts about float mechanisms past and present, together with a summary of various features, some unresolved:

1. The value **\textfraction** does not affect the processing of doublecol floats: this seems sensible, but should be documented.
2. **\twocolumn** floatplacement was wrong: dbl not needed, ord needed.
3. **\@floatplacement** was not called after **\@startdblcol** or **\@topnewpage**. This has been changed; it is clearly a bug fix.
4. The use **\@topnewpage** when **\dblfigrule** is non-trivial produced a rule in the wrong place. This has been fixed by not using **\dblfigrule** when processing the 'float' from **\@topnewpage**.

5. If the specifier was just h and the float could not be put here, it went on the deferlist and stayed there until a clearpage. It now gets changed to a 'th': this is only an error-recovery action, putting just h or !h should be deprecated.

6. `\@dblmaxsep` was 'the maximum of `\dblfloatsep` and `\dbltexfloatsep`'. But it was never used! Now gone completely, like `\@maxsep`.

7. After an h float is put on a page, it was counted as text when applying the `\textfraction` test; this is possibly too big a change although it is a bug fix?

8. Two consecutive h floats are separated by twice `\intextsep`: this could be changed to one by use of `\addvspace`, OK? Note that it would also mean that less space is put in if an h float immediaiely follows other spaces. This is also possibly too big a change, at least for compatibility mode? Or it may be simply wrong! It has not been changed.

9. Now `\@addtocurcol` checks first for just p fps. I think that this is an increase in efficiency, but maybe the coding should be made even more efficient.

10. `\@tryfcolumn` now tests if the list is empty first, otherwise lots of wasted time! Thus this test has been removed from `\@startcolumn`. As Frank pointed out, this makes `\@startcolumn` less efficient. But it is now the same as `\@startdblcolumn`: I can see no reason why they should be different, but which is best?

11. Why is `\@colroom` set in `\@doclearpage`?

12. Should `\@maxdepth` store the setting of `\maxdepth` from lplain? Or should we provide a proper interface to class files for setting these?

```
\def\@makefcolumn #1{%
  \begingroup
    \@fpmin \z@
    \let \@testfp \@gobble
    \@tryfcolumn #1%
  \endgroup
}
```

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. [5]

```
\def\@outputdblcol{%
  \if@firstcolumn
    \global \@firstcolumnfalse
    \global \setbox\@leftcolumn \box\@outputbox
%<*trace>
    \tr@ce{PAGE: first column boxed}%
```

[5] 1.2g 2000/07/12 Ensure that rule is in `\normalcolor`

```
%</trace>
  \else
    \global \@firstcolumntrue
    \setbox\@outputbox \vbox {%
                          \hb@xt@\textwidth {%
                            \hb@xt@\columnwidth {%
                              \box\@leftcolumn \hss}%
                            \hfil
                            {\normalcolor\vrule ⤸
                                 ↊\@width\columnseprule}%
                            \hfil
                            \hb@xt@\columnwidth {%
                              \box\@outputbox \hss}%
                                              }%
                             }%
%<*trace>
    \tr@ce{PAGE: second column also boxed}%
%</trace>
    \@combinedblfloats
    \@outputpage
%<*trace>
    \tr@ce{PAGE: two column page completed}%
%</trace>
    \begingroup
      \@dblfloatplacement
      \@startdblcolumn
      \@whilesw\if@fcolmade \fi
        {\@outputpage
%<*trace>
      \tr@ce{PAGE: double float page completed}%
%</trace>
          \@startdblcolumn}%
    \endgroup
  \fi
}
```

39.18.00 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. Unlike many other parameters that LaTeX 2_ε leaves to be determined by the classes default values are entered here.

39.18.00 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.

```
\newcount\c@topnumber
\setcounter{topnumber}{2}
```

\topfraction

This macro holds the maximum proportion (as a decimal number) of a text page or column that can be occupied by floats at the top. In the **phd** package, we set this as **.85**. Many of the images we use are quite high and a value of **.85** is more appropriate.

```
\newcommand\topfraction{.7}
```

\c@bottomnumber

This counter holds the maximum number of floats that can appear at the bottom of a text page or column. Lamport's value to allow only one is reasonable for many publications, but fails in many others. In the **phd** package we allowed a much larger number as we are aiming for more compact documents.

```
\newcount\c@bottomnumber
\setcounter{bottomnumber}{1}
```

\bottomfraction

This macro holds the maximum proportion (as a decimal number) of a text page or column that can be occupied by floats at the bottom.

```
\newcommand\bottomfraction{.3}
```

\c@totalnumber

This counter holds the maximum number of floats that can appear on any text page or column.

```
\newcount\c@totalnumber
\setcounter{totalnumber}{3}
```

\textfraction

This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text.

```
\newcommand\textfraction{.2}
```

\floatpagefraction

This macro holds the minimum proportion (as a decimal number) of a page or column that must be occupied by floating objects before a 'float page' is produced.

```
\newcommand\floatpagefraction{.5}
```

\c@dbltopnumber

This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page.

```
\newcount\c@dbltopnumber
\setcounter{dbltopnumber}{2}
```

\dbltopfraction

This macro holds the maximum proportion (as a decimal number) of a two-column text page that can be occupied by double-column floats at the top.

```
\newcommand\dbltopfraction{.7}
```

\dblfloatpagefraction

This macro holds the minimum proportion (as a decimal number) of a page that must be occupied by double-column floating objects before a 'double-column float page' is produced.

```
\newcommand\dblfloatpagefraction{.5}
```

39.18.1 Floats on a text page

\floatsep

\textfloatsep

\intextsep

When a floating object is placed on a page with text, these parameters control the seperation between the float and the other objects on the page. These parameters are used for both one-column mode and single-column floats in two-column mode. They are all rubber lengths.

\floatsep is the space between adjacent floats that are placed at the top or bottom of the text page or column.

\textfloatsep is the space between the main text and floats at the top or bottom of the page or column.

\intextsep is the space between in-text floats and the text.

```
\newskip\floatsep
\newskip\textfloatsep
\newskip\intextsep
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
```

\dblfloatsep

\dbltextfloatsep

When double-column floats (floating objects that span the whole **\textwidth**) are placed at the top of a text page in two-column mode, the separation between the float and the text is controlled by **\dblfloatsep** and **\dbltextfloatsep**. They are rubber lengths.

\dblfloatsep is the space between adjacent double-column floats placed at the top of the text page.

\dbltextfloatsep is the space between the main text and double-column floats at the top of the page.

```
\newskip\dblfloatsep
\newskip\dbltextfloatsep
\setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

39.18.2 Floats on their own page or column

When floating objects are placed on a seperate page or column this is called a 'float page', the layout of the page is controlled by these parameters, which are rubber lengths.

`\@fptop`

`\@fpsep`

`\@fpbot`

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.

```
\newskip\@fptop
\newskip\@fpsep
\newskip\@fpbot
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblfptop`

```
  \newskip\@dblfptop
```

`\@dblfpsep`

`\@dblfpbot`

Double-column 'float pages' in two-column mode use similar parameters.

```

\newskip\@dblfpsep
\newskip\@dblfpbot
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
```

`\topfigrule`

`\botfigrule`

`\dblfigrule`

The macros can be used to put in rules between floats and text; whatever they insert should be vertical mode material which takes up zero space.

```
\let\topfigrule=\relax
\let\botfigrule=\relax
\let\dblfigrule=\relax
```

phd package and give a background as to how fonts are used in LaTeX.

First there was one user and I took a lot of time to satisfy myself. Then I had 10 users, and a whole new level of difficulties arose. Then I had a hundred users and another level of things happened. I had a thousand users, I had ten thousand each of those were special phases in the development, important. I couldn't have gone with ten thousand until I'd done it with a thousand. But each time a new wave of changes came along, the idea was to have TEX get better, and not get more diverse as it needed to handle new things.

Donald Knuth

40.1 Introduction

To *make* a book is an interesting and somewhat involved process[1]. The text is set in type and printed on pages, the pages are gathered and folded into signatures and these are gathered and folded into signatures and these are then bound and covered. Many of the aspects of this process that has passed down to us by previous generations is discussed extensively in other sections of this book. Class authors have to distill this knowledge in a set of typographical rules to be described in a class file. The first thing such an author must do is to describe the *rationale* of developing such a class. The **octavo**[2] class was developed to enable printing books in dimensions that follow traditional styles. The[3] class to offer a flexible system on which other classes could be based and so does[4]. The `tufte-book` and `tufte-handout` classes to provide a style that resembles those found in Tufte books. Many Universities offer *Thesis* classes to standardize the way these are produced. Many of these Universities, translated the styles previously typed and the results are a typographical disaster, only mitigated by the ability to display beautiful mathematics. As these are printed on standard *photocopy paper* one cannot do much with the layout.

40.2 What is a class?

A class is simply a file with the extension `cls` containg a set of macros. A class can load another class.

40.3 Identifying your class

The first thing a class must do is to identify any other formats it needs and to announce its name. This is accomplished using the two commands **??**[→ P. ??] and **??**[→ P. ??].

The following example, delares the version of LaTeX 2_ε that it requires and then gives the class name. It can be found in the preable of most well written

[1] **town**.
[2] **octavo**.
[3] **memoir**.
[4] **koma**.

classes. You should also put some remarks to identify you as the author, the version number and other similar details. These are discussed in more detail in the next Chapter, where you will see how to automate documentation for your class.

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesClass{myclass-book}[2010/12/11 v3.5.0 myclass-book]
```

The above syntax must be followed exactly so that this information can be used by `LoadClass` or `documentclass` (for classes) or `\RequirePackage` or`\sepackage` (for packages) to test that the release is not too old. The whole of this $< release - info >$ information is displayed by `\listfiles` and should therefore not be too long.

```
% Load the common style elements
\input{myclass-common.def}
```

Another command that can be used is `\ProvidesFile`. This is similar to the two previous commands except that here the fullname, including the extension, must be given. It is used for declaring any files other than main class and package files.

This is useful, if you decide to have your main definitions in a separate file.

40.4 Class Options

Before we see in detail how to add options to a class, we need to review a package called **xkeyval**

. Unless you are in the business of re-discovering wheels, this is an absolute must for developing, readable and maintenable code and your class is to provide many options.

```
\usepackage[textcolor=red,font=times]{mypack}
```

Class options are best set by using booleans`\newboolean`.

We first set a new boolean that we `name@myclass@afourpaper.` This is used using the package `ifthen`[5] Then we can `DecalareOptionX` and we set the boolean to default to true. If the user then types

myclass[a4paper]

The a4paper options will be set. This is a much better and concise way of defining options. `\ewboolean`

```
\newboolean{@myclass@afourpaper}
\DeclareOptionX[myclass]<common>{a4paper}
  {
   \setboolean{@myclass@afourpaper}
   {true}
  }
```

Note that the command provide by `ifthen` `\setboolean` takes true or false, as #2, and sets #1 accordingly. In the above code we set the option as true.

It is much easier and most programmers use the `ifthen` package to check for option booleans

```
\ifthenelse{\boolean{@myclass@afourpaper}}
```

[5]The ifthen package was developed by David Carlisle, can be downloaded at http://www.ifi.uio.no/it/latex-links/ifthen.pdf

```
{\geometry{
      a4paper,
      left=24.8mm,
      top=27.4mm,
      headsep=2\baselineskip,
      textwidth=107mm,
      marginparsep=8.2mm,
      marginparwidth=49.4mm,
      textheight=49\baselineskip,
      headheight=\baselineskip
   }
 }
{}
```

40.5 Set-up the font sizes

LaTeX does not provide definitions of all the font-sizes. Unless you are extending an existing class, this is one of the first tasks you need to do in your new class.

Normally class authors will define all the commonly defined size commands, such as `\mall`, `\ormalsize` and other similar commands.

In the example shown below, we first start by defining the `\ormalsize` font size. In this book the `\normalsize` is defined as 14pt. We also define the vertical spaces that we need to have abovedisplay and belowdisplayskip. These are all very difficult to remember and once you have something you are happy with, just copy from class to class or even define a samll definition file to keep them all together.

Helvetica looks like this and Palatino looks like this.

The user has access to a number of commands which change the size of the fount, relative to the 'main' size used for the bulk of the text.

These `\ize` commands issue a `\setfontsize` command.

```
\@setfontsize\size\font-size{baselineskip} where:
```

font-size The absolute size of the fount to use from now on.

baselineskip The normal value of `\aselineskip` for the size of the fount selected. (The actual value will be

A number of commands, defined in the LaTeXkernel, shorten the following definitions and are used throughout. These are:

`\@vpt`	5	`\@vipt`	6	`\@viipt`	7
`\@viiipt`	8	`\@ixpt`	9	`\@xpt`	10
`\@xipt`	10.95	`\@xiipt`	12	`\@xivpt`	14.4

...

40.5.1 Setting up the normalsize

The user command to obtain the 'main' size is `\ormalsize`. LaTeX uses `\normalsize` when referring to the main size and maintains this

value even if \normalsize is redefined. The \normalsize macro also sets values for \bovedisplayskip, \bovedisplayshortskip and \elowdisplayshortskip.

```
%%
% Set the font sizes and baselines to match Tufte's books
% normalsize
%%
\renewcommand\normalsize{%
   \@setfontsize\normalsize\@xpt{14}%
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \belowdisplayskip \abovedisplayskip
   \let\@listi\@listI}

\normalbaselineskip=14pt
\normalsize


\renewcommand\small{%
   \@setfontsize\small\@ixpt{12}%
   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus2\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip
}
\renewcommand\footnotesize{%
   \@setfontsize\footnotesize\@viiipt{10}%
   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus\p@
   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 3\p@ \@plus\p@ \@minus\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip
}
\renewcommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\renewcommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\renewcommand\large{\@setfontsize\large\@xipt{15}}
\renewcommand\Large{\@setfontsize\Large\@xiipt{16}}
\renewcommand\LARGE{\@setfontsize\LARGE\@xivpt{18}}
\renewcommand\huge{\@setfontsize\huge\@xxpt{30}}
\renewcommand\Huge{\@setfontsize\Huge{24}{36}}

%% Define a HUGE for fun
\newcommand\HUGE{\@setfontsize\Huge{38}{47}}
```

40.6 Adjusting paragraph parameters

The parameters which control TEX's behaviour when typesetting paragraphs receive a bit of a tweak here. Contrary to the usual behaviour of modifying the grid with glue when difficulties are encountered with vertical space, here we shall try to counteract these tendencies and enforce as much as possible uniformity of the grid of lines.

A good value for paragraph indentation is `parindent 0.5pt`, for vertical spacing between paragraphs that are indented use opt. At this point if you are using any marginals it is a good idea to allow hyphenation with the `ragged2e` package. Since marginals use very narrow paragraphs you may get a very funny looking marginal text. Using the package, adjustments can be made to hyphenate the marginal text.

```
%%
% \RaggedRight allows hyphenation

\RequirePackage{ragged2e}
\setlength{\RaggedRightRightskip}{\z@ plus 0.08\hsize}
\setlength{\RaggedRightParindent}{1pc}

% Paragraph indentation and separation for normal text
\newcommand{\@tufte@reset@par}{%
  \setlength{\RaggedRightParindent}{1.0pc}%
  \setlength{\parindent}{1pc}%
  \setlength{\parskip}{0pt}%
}
\@tufte@reset@par

% Paragraph indentation and separation for marginal text
\newcommand{\@tufte@margin@par}{%
  \setlength{\RaggedRightParindent}{0.5pc}%
  \setlength{\parindent}{0.5pc}%
  \setlength{\parskip}{0pt}%
}
```

40.7 Formatting Chapters and Sections

The section on Chapters etc, has more on this, but we will touch on it briefly. Most recent class developerss use the **titlesec** and **titletoc** package to handle the complexity of these commands. With the **phd** package this is unecessary.

```
\titleformat{\subsection}%
  [hang]% shape
  {\normalfont\large}% format applied to label+text removed \itshape
  {\thesubsection}% label
```

```
  {1em}% horizontal separation between label and title body
  {}% before the title body
  []% after the title body
```

These are normally followed by the “titlespacing” commands to define the space around these sections.

```
%% We set the titlespacing using the package titlesec and titletoc
%
\titlespacing*{\chapter}{0pt}{20pt}{40pt}
\titlespacing*{\section}{0pt}{3.5ex plus 1ex minus .2ex}{2.3ex plus .2ex}
\titlespacing*{\subsection}{0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus.2ex}
```

40.8 Adjusting the Index

For classes representing books, the index is treated like a chapter whereas for others it is normally treated like a section. Whatever your document ends up like, indices are best done in a multi-column environment. One possibility is shown below, using the package ”multcol”.

```
\RequirePackage{multicol}
\renewenvironment{theindex}
  {\begin{fullwidth}%
    \small%
    \ifthenelse{\equal{\@tufte@class}{book}}%
      {\chapter{\indexname}}%
      {\section*{\indexname}}%
    \parskip0pt%
    \parindent0pt%
    \let\item\@idxitem%
    \begin{multicols}{3}%
  }
  {\end{multicols}%

\renewcommand\@idxitem{\par\hangindent 2em}
\renewcommand\subitem{\par\hangindent 3em\hspace*{1em}}
\renewcommand\subsubitem{
    \par\hangindent 4em\hspace*{2em}
}
\renewcommand\indexspace{
    \par\addvspace{
       1.0\baselineskip plus 0.5ex minus 0.2ex}\relax
    }%
%we now  swallow the letter heading in the index
\newcommand{\lettergroup}[1]{}
```

The code, renews the ”theindex” environment, with minor tweaks and defines it as a three column layout at ”fullwidth”.

40.9 Provide some hooks

It is useful at the end of the class to allow for localization of the class by importing a local file. This is easily achieved by checking if the file exists and then loading it. If there is a **myclass-book-local.sty** file, load it.

```
\IfFileExists{myclass-book-local.tex}
  {input{myclass-book-local}
   \MyClassInfoNL{Loading myclass-book-local.tex}}
  {}
```

If you intent to publish your class, you may also want to consider adding a hook for a patch-file.

40.10 The final act of kindness to your users

Many common classes, such as the **memoir** use such a tactic to avoid breaking old code.

```
 \IfFileExists{mypatch.sty}{%
 \RequirePackage{mypatch}}{}
```

phd package and give a background as to how fonts are used in LaTeX.

In the previous chapter we have outlined the main sections that you probably need to define in your class. In the examples we have used we just typed the examples as **example.cls** or **package.sty**.

In this chapter we will go over the packaging of the class and automating the generation of user documentation, using the **doc** and **DocStrip**[1] programs in files with an extension **.dtx**. The DocStrip program is an amazing piece of code that was originally created by Frank Mittelbach to accompany the **doc** package. The idea behind it was to remove comment lines in order to reduce the execution time of the program. Having created the DocStrip program to remove comment lines from programs it became feasible to do more than just strip comments. Wouldn't it be nice to have a way to include parts of the code only when some condition is set true? Wouldn't it be as nice to have the possibility to split the source of a TEX program into several smaller files and combine them later into one 'executable'? Both these wishes have been implemented in the DocStrip program.

You should also be familiar with "LaTeX2e" for Class and Package Writers", which is available from CTAN (http://www.ctan.org) and comes with most LaTeX2e" distributions in a file called clsguide.dvi.[2] Finally, you should know how to install packages that are shipped as a .dtx file plus a .ins file.

style (.sty) file is primarily a collection of macro and environment definitions. One or more style files (e.g., a main style file that **\input** or **\RequirePackages** multiple helper files) is called a package. Packages are loaded into a document with **\usepackage**{⟨*main .sty fille*⟩}. In the rest of this document, we use the notation ⟨*package*⟩ to represent the name of your package.

Motivation The important parts of a package are the code, the documentation of the code, and the user documentation. Using the Doc and DocStrip programs, it's possible to combine all three of these into a single, documented LATEX(.dtx) file. The primary advantage of a .dtx file is that it enables you to use arbitrary LATEX constructs to comment your code. Hence, macros, environments, code stanzas, variables, and so forth can be explained using tables, figures, mathematics, and font changes. Code can be organized into sections using LATEX's sectioning commands. Doc even facilitates generating a unified index that indexes both macro definitions (in the LATEX code) and macro descriptions (in the user documentation).

This emphasis on writing verbose, nicely typeset comments for code—essentially treating a program as a book that describes a set of algorithms—is known as literate programming **literate** and has been in use since the early days of TEX .

Furthermore, this tutorial shows how to write a single file that serves as both documentation and driver file, which is a more typical usage of the Doc system than using separate files.

41.0.1 The .ins file

The first step in preparing a package for distribution is to write an installer (**.ins**) file. An installer file extracts the code from a **.dtx** file, uses **docstrip** to

[1] Mittelbach et al. (2018).
[2] **pakin2004**.

strip off the comments and documentation, and outputs a `.sty` file. The good news is that a `.ins` file is typically fairly short and doesn't change significantly from one package to another.

License The `ins` files usually start with comments specifying the copyright and license information:

```
%%
%% Copyright (C) year by your name %%
%% This file may be distributed and/or modified under the
%% conditions of the LaTeX Project Public License, either
%% version 1.2 of this license or (at your option) any later
%% version. The latest version of this license is in:
%%
%% http://www.latex-project.org/lppl.txt
%%
%% and version 1.2 or later is part of all distributions of
%% LaTeX version 1999/12/01 or later.
%%
```

The LATEX Project Public License (LPPL) is the license under which most packages—and LATEX itself—are distributed. Of course, you can release your package under any license you want; the LPPL is merely the most common license for LATEX packages. The LPPL specifies that a user can do whatever he wants with your package—including sell it and give you nothing in return. The only restrictions are that he must give you credit for your work, and he must change the name of the package if he modifies anything to avoid versioning confusion. The next step is to load DocStrip:

```
%%\input docstrip.tex
%%\keepsilent
```

By default, DocStrip gives a line-by-line account of its activity. These messages aren't terribly useful, so most people turn them off, by using the command `\keepsilent`:

```
\keepsilent
```

A system administrator can specify the base directory under which all TEX-related files should be installed, e.g., `/usr/share/texmf`. (See `\BaseDirectory` in the DocStrip manual.) The `ins` file specifies where its files should be installed relative to that. The following is typical:

```
\usedir{tex/latex/packagename}
\preamble
htexti \endpreamble
```

The next step is to specify a preamble, which is a block of commentary that will be written to the top of every generated file:

```
\preamble
----------------------------------------------------------------
phddoc --- A class to typeset LaTeX code.
```

```
E-mail: yannislaz@gmail.com
Released under the LaTeX Project Public License v1.3c or later
See http://www.latex-project.org/lppl.txt
----------------------------------------------------------------
\endpreamble
```

The preceding preamble would cause **package.sty** to begin as follows:

```
%%
%% This is file `phddoc.cls',
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% phddoc.dtx  (with options: `class')
%% ----------------------------------------------------------------
%% phddoc --- A class to typeset LaTeX code.
%% E-mail: yannislaz@gmail.com
%% Released under the LaTeX Project Public License v1.3c or later
%% See http://www.latex-project.org/lppl.txt
%% ----------------------------------------------------------------
```

We now reach the most important part of a .ins file: the specification of what files to generate from the **.dtx** file. The following tells DocStrip to generate hpackagei.sty from hpackagei.dtx by extracting only those parts marked as 'package' in the .dtx file. (Marking parts of a .dtx file is described later on.)

```
\generate{\file{<package>.sty}{\from{<package>.dtx}{package}}}
```

\generate can extract any number of files from a given .dtx file. It can even extract a single file from multiple **.dtx** files. See the DocStrip manual for details.

Personally I also generate README.md files in **markdown** format as well, so that when they get uploaded to **github** they can be rendered nicely.

```
\generate{\file{\jobname.md}{\from{\jobname.dtx}{readmemd}}}
```

The text has to be wriiten using 'guards' with the tag **readmd**

```
%<*readmemd>
# The `phddoc` LaTeX2e class

The `phd` latex package and the class with the same name provide
convenient methods to create new styles for books, reports
and articles. It also loads the most commonly used packages
and resolves conflicts.
%</readmemd>
```

41.0.2 Generating messages

The next part of a **.ins** file consists of commands to output a message to the user, telling him what files need to be installed and reminding him how to produce the user documentation. The following set of **\sg** commands is typical:

```
\obeyspaces
\Msg{****************************************************}
\Msg{* *}
\Msg{* To finish the installation you have to move the *}
\Msg{* following file into a directory searched by TeX: *}
\Msg{* *}
\Msg{* packagei.sty *}
\Msg{* *}
\Msg{* To produce the documentation run the file *}
\Msg{* package.dtx through LaTeX. *}
\Msg{* *}
\Msg{* Happy TeXing! *}
\Msg{* *}
\Msg{****************************************************}
Note the use of \obeyspaces to inhibit \tex from collapsing multiple spaces
into one.
\endbatchfile
```

Appendix A.1 lists a complete, skeleton .ins file. Appendix A.2 is similar but contains slight modifications intended to produce a class (`.cls`) file instead of a style (`.sty`) file

41.1 What to put in a .dtx file

We started describing the `.ins` install file first. The next file we will describe is the `.dtx` file. This holds both the code definitions as well as the user documentation.

A `dtx` file contains both the commented source code and the user documentation for the package. Running a `dtx` file through `latex` typesets the user documentation, which usually also includes a nicely typeset version of the commented source code.

Due to some Doc trickery, a `dtx` file is actually evaluated twice. The first time, only a small piece of LaTeX driver code is evaluated. The second time, comments in the `dtx` file are evaluated, as if there were no '%' preceding them. This can lead to a good deal of confusion when writing `dtx` files and occasionally leads to some awkward constructions. Fortunately, once the basic structure of a `dtx` file is in place, filling in the code is fairly straightforward.

Guards If you open any .dtx file you will notice that the lines either start with a % sign or sometimes with a percentage sign and <*guard*>. The latter is called a guard and they are in a way like html tags. They have a starting and an ending tag. In the example below there are two different guards `<*10pt>...</10pt>` and `<*11pt></11pt>`. Unlike html tags guards are boolean expressions! You can use:

|! &

The |stands for disjunction (OR), the & stands for conjunction (AND) and the ! (NOT) stands for negation. The terminal is any sequence of letters and evaluates to true iff it occurs in the list of options that have to be included.

```
%<*10pt|11pt|12pt>
... code
%</10pt|11pt|12pt>
```

A longer example from KOMA shows the concept better.

```
%    \begin{macrocode}
\def\normalsize{%
%<*10pt>
  \@setfontsize\normalsize\@xpt\@xiipt
  \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
  \abovedisplayshortskip \z@ \@plus3\p@
  \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
%</10pt>
%<*11pt>
  \@setfontsize\normalsize\@xipt{13.6}%
  \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
  \abovedisplayshortskip \z@ \@plus3\p@
  \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
%</11pt>
...
%    end{macrocode}
```

If the guards only contain a one line of text, then a short form is provided as `<10pt>`. It is unecessary to provide a closing tag and the '*' is omitted. The example below from the KOMA classes shows a quite ingenious way of writing the `\ProvidesFile` macro in the different files; one for each tag. Two kinds of optional code are supported: one can either have optional code that is on one line of tex code.

To distinguish both kinds of optional code the 'guard modi

er' has been introduced. The 'guard modifier' is one character that immediately follows the < of the guard. It can be either * for the beginning of a block of code, or / for the end of a block of code. The beginning and ending guards for a block of code have to be on a line by themselves.

When a block of code is not included, any guards that occur within that block are not evaluated.

```
%    \begin{macrocode}
\ProvidesFile{%
%<10pt>  scrsize10pt.clo%
%<11pt>  scrsize11pt.clo%
%<12pt>  scrsize12pt.clo%
}[\KOMAScriptVersion\space font size class option %
%<10pt>  (10pt)%
%<11pt>  (11pt)%
%<12pt>  (12pt)%
]
%    \end{macrocode}
```

In the `.ins` file one could write to generate the various `.clo` files.:

```
\generate{\usepreamble\defaultpreamble
  \file{scrsize10pt.clo}{%
    \from{scrkernel-version.dtx}{clo,10pt}%
    \from{scrkernel-fonts.dtx}{clo,10pt}%
    \from{scrkernel-paragraphs.dtx}{clo,10pt}%
  }%
  \file{scrsize11pt.clo}{%
    \from{scrkernel-version.dtx}{clo,11pt}%
    \from{scrkernel-fonts.dtx}{clo,11pt}%
    \from{scrkernel-paragraphs.dtx}{clo,11pt}%
  }%
  \file{scrsize12pt.clo}{%
    \from{scrkernel-version.dtx}{clo,12pt}%
    \from{scrkernel-fonts.dtx}{clo,12pt}%
    \from{scrkernel-paragraphs.dtx}{clo,12pt}%
  }%
}%
```

Becareful not to introduce spurious empy lines in your generated files by having empty lines in no-man's land, that is between tags.[3]

```
%</install>

%<install>\endbatchfile
```

The character table check The second mechanism that Doc uses to ensure that a `dtx` file is uncorrupted is a character table. If you put the following command verbatim into your `dtx` file, then **Doc** will ensure that no unexpected character translation took place in transport:

```
% \CharacterTable
% {Upper-case \A\B\C\D\E\F\G\H\I\J\K\L\M\N\O\P\Q\R\S\T\U\V\W\X\Y\Z
% Lower-case \a\b\c\d\e\f\g\h\i\j\k\l\m\n\o\p\q\r\s\t\u\v\w\x\y\z
% Digits \0\1\2\3\4\5\6\7\8\9
% Exclamation \! Double quote \" Hash (number) \#
% Dollar \$ Percent \% Ampersand \&
% Acute accent \' Left paren \( Right paren \)
% Asterisk \* Plus \+ Comma \,
% Minus \- Point \. Solidus \/
% Colon \: Semicolon \; Less than \<
% Equals \= Greater than \> Question mark \?
% Commercial at \@ Left bracket \[ Backslash \\
% Right bracket \] Circumflex \^ Underscore \_
% Grave accent \` Left brace \{ Vertical bar \|
% Right brace \} Tilde \~}
A success message looks like this:
***************************
* Character table correct *
```

[3] In the phd package, I automatically generate the default settings from the `.dtx` files. In this case pgf will complain.

```
**************************

and an error message looks like this:
! Package doc Error: Character table corrupted.
```

DoNotIndex When producing an index, **doc** normally indexes every control sequence (i.e., backslashed word or symbol) in the code. The problem with this level of automation is that many control sequences are uninteresting from the perspective of understanding the code. For example, a reader probably doesn't want to see every location where `\if` is used—or `\the` or `\let` or `\begin` or any of numerous other control sequences.

As its name implies, the `\DoNotIndex` command gives `Doc` a list of control sequences that should not be indexed. `\DoNotIndex` can be used any number of times, and it accepts any number of control sequence names per invocation:

```
\DoNotIndex{\#,\$,\%,\&,\@,\\,\{,\},\^,\_,\~,\ }
\DoNotIndex{\@ne}
\DoNotIndex{\advance,\begingroup,\catcode,\closein}
\DoNotIndex{\closeout,\day,\def,\edef,\else,\empty, \endgroup}
```

41.1.1 User documentation

We can finally start writing the user documentation. A typical beginning looks like this:

```
% \title{The \textsf{package} package\thanks{This document
% corresponds to \textsf{package}~\fileversion,
% dated~\filedate.}}
% \author{your name \\ \texttt{your e-mail address}}
%
% \maketitle
```

The title can certainly be more creative, but note that it's common for package names to be typeset with `\textsf` and for `\thanks` to be used to specify the package version and date. This yields one of the advantages of literate programming: Whenever you change the package version (the optional second argument to `\ProvidesPackage`), the user documentation is updated accordingly. Of course, you still have to ensure manually that the user documentation accurately describes the updated package.

Write the user documentation as you would any LaTeX document, except that you have to precede each line with a `\%`. Note that the `ltxdoc` document class is derived from article, so the top-level sectioning command is `\section`, not `\chapter`.

41.2 General tips for defining a Class

Evaluate, if there is a class that is nearer to what you wish to achive. If not do a set of requirements.

Book structure - start with book or **Octavo** if you need to hack extensively. If not use memoir, **koma** or **tufte-book**.

Paragraph looks
Lists
Figures
Bibliography and citations
Footnotes
Index
Title pages
Book Cover
Language support
Mathematics
Graphs and figures
Typography - fonts, indentations fontsize etc
headers and footers

41.3 Declaring Options

Most classes or packages will have a good deal of options. These are declared using the **\DeclareOption** command. In this part no package loading should take place.

\DeclareOption {⟨*option*⟩} {⟨*code*⟩}

The argument option is the name of the option being declared and the {⟨*code*⟩} is the code that will execute if this option is requested.

\DeclareOption* {⟨*code*⟩}

The argument ⟨*code*⟩ in the star version of the command specifies the action to be taken if an unknown option is specified. Within this argument the **\CurrentOption** refers to the name of the option in question.

For example one could pass all such options to another package, using:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{A}}
```

41.4 Executing Options

Normally after the options have been defined, one would need to provide default values and the options need to be executed.

\ExecuteOptions {⟨*option list*⟩}

You can also **\ExecuteOptions** when declaring other options. There is one caveat. This command can only be executed prior to executing the

`\ProcessOptions` command because, as one of its last actions, the latter command reclaims all of the memory taken up by the code for the declared options.

`\ProcessOptions*`

For some packages it is preferable or essential to process options in the order they appear in the `usepackage` commands rather than using the order given through the sequence of the `\DeclareOption` commands. In this case it one has to use the star version of the command, i.e, `\ProcessOptions*` rather than `\ProcessOptions`.

41.5 Special Commands for class files

It is sometimes preferable to define a new class based on another and hence to extend it. To support this concept the LaTeX kernel provides two commands, `\LoadClass` and `\PassOptionsToClass`. These two commands can then be used to develop a new class, by adding and extending the functionality of the loaded class.

?? → P. ?? [⟨*option list*⟩]{⟨*class*⟩}[⟨*release*⟩]

For example the `ltxdoc` class loads the standard `article` class. The **tufte-book** class loads the `book` class. The best way to understand the concepts discussed here is to study these classes.

41.6 A minimal class

Example 6.1 Model Class

```

\begin{filecontents}{phdexampleclass.cls}
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{phdexampleclass}[2015/07/07]
\renewcommand\normalsize{\fontsize{}{10pt}{12pt}\selectfont}
\setlength\textwidth{6.5in}
\setlength\textheight{5in}
\pagenumbering{arabic}
\end{filecontents}
```

phd package and give a background as to how fonts are used in LaTeX.

42.1 Introduction

This part of the kernel implements the class and package mechanisms, including the handling of options. It replaced the LaTeX 2.09 **\documentstyle** in LaTeX 2_ε documents. Provisions were made for backwards compatibility so that old documents containing **\documentstyle** could still be run using a compatibility option. Backwards compatibility has always been one of the primary objectives of the LaTeX Team making LaTeX an excellent resource for archival purposes.

The LaTeX kernel on its own does not provide the necessary functionality to typeset a document. It is a framework upon which classes can define the necessary elements and the instructions as to how they are formatted. Packages can then be used to extend even further the functionality of the classes or even the kernel. Classes and packages can be loaded with options.

42.2 User interface

\documentclass[⟨*main-option-list*⟩]{⟨*class*⟩}[⟨*version*⟩]

There must be exactly one such declaration, and it must come first. The ⟨*main-option-list*⟩ is a list of options which can modify the formatting of elements which are defined in the ⟨*class*⟩ file as well as in all following **\usepackage** declarations (see below). The ⟨*version*⟩ is a version number, beginning with a date in the format **YYYY/MM/DD**. If an older version of the class is found, a warning is issued.

\documentstyle[⟨*main-option-list*⟩]{⟨*class*⟩}[⟨*version*⟩]

The **\documentstyle** declaration is kept in order to maintain upward compatibility with LaTeX2.09 documents. It is similar to **\documentclass**, but it causes all options in ⟨*main-option-list*⟩ that the ⟨*class*⟩ does not use to be passed to **\RequirePackage** after the options have been processed. This maintains compatibility with the 2.09 behaviour. Also a flag is set to indicate that the document is to be processed in LaTeX2.09 compatibility mode. As far as most packages are concerned, this only affects the warnings and errors LaTeX generates. This flag does affect the definition of font commands, and **\sloppy**.

\usepackage[⟨*package-option-list*⟩]{⟨*package-list*⟩}[⟨*version*⟩]

There can be any number of these declarations. All packages in ⟨*package-list*⟩ are called with the same options.

Each ⟨*package*⟩ file defines new elements (or modifies those defined in the ⟨*class*⟩), and thus extends the range of documents which can be processed.

The ⟨*package-option-list*⟩ is a list of options which can modify the formatting of elements defined in the ⟨*package*⟩ file. The ⟨*version*⟩ is a version number, beginning with a date in the format **YYYY/MM/DD**. If an older version of the package is found, a warning is issued.

Each package is loaded only once. If the same package is requested more than once, nothing happens, unless the package has been requested with options that were not given the first time it was loaded, in which case an error is produced.

As well as processing the options given in the ⟨*package-option-list*⟩, each package processes the ⟨*main-option-list*⟩. This means that options that affect

all of the packages can be given globally, rather than repeated for every package.

Note that class files have the extension `.cls`, packages have the extension `.sty`.

The environment **??**[→P.??] is intended for passing the contents of packages, options, or other files along with a document in a single file.

It has one argument, which is the name of the file to create. If that file already exists (maybe only in the current directory if the OS supports a notion of a 'current directory' or 'default directory') then nothing happens (except for an information message) and the body of the environment is bypassed. Otherwise, the body of the environment is written verbatim to the file name given as the first argument, together with some comments about how it was produced.

The environment is allowed only before **??**[→P.??] to ensure that all packages or options necessary for this particular run are present when needed. The begin and end tags should each be on a line by itself. There is also a star-form; this does not write extra comments into the file.

42.2.1 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 **??**[→P.??] 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.

42.3 Class and Package interface

42.3.1 Class name and version

A class can identify itself with the **??**[→ P. ??]`{`⟨*name*⟩`}[`⟨*version*⟩`]` command. The ⟨*version*⟩ should begin with a date in the format `YYYY/MM/DD`.

42.3.2 Package name and version

A package can identify itself with the **??**[→ P. ??]`{`⟨*name*⟩`}[`⟨*version*⟩`]` command. The ⟨*version*⟩ should begin with a date in the format `YYYY/MM/DD`.

42.3.3 Requiring other packages

Packages or classes can load other packages using
??[→ P. ??]`[`⟨*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.

??[→ P. ??] is similar to `\RequirePackage`, but for classes, may not be used in package files.

`\PassOptionsToPackage` Packages can pass options to other packages using:
`\PassOptionsToPackage{`⟨*options*⟩`}{`⟨*package*⟩`}`.

`\PassOptionsToClass` This adds the ⟨*options*⟩ to the options list of any future `\RequirePackage` or `\usepackage` command. For example:

```
\PassOptionsToPackage{foo,bar}{fred}
\RequirePackage[baz]{fred}
```

is the same as:

```
\RequirePackage[foo,bar,baz]{fred}
```

`\LoadClassWithOptions{`⟨*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`.

??[→ P. ??] 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.

??→P.?? To find out if a package has already been loaded.

??→P.?? To find out if a class has already been loaded.

??→P.??`{`⟨*package*⟩`}{`⟨⟨*true*⟩⟩`}{`⟨⟨*false*⟩⟩`}` To find out if a package has been loaded.

??→P.?? [1] To find out if a package has already been loaded with a version equal to or more recent than ⟨*version*⟩.

??→P.?? `\@ifpackagelater{`⟨*package*⟩`}{`⟨*version*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

??→P.?? To find out if a package has already been loaded with at least the options ⟨*options*⟩, use

??→P.?? `\@ifpackagewith{`⟨*package*⟩`}{`⟨*options*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

There exists one package that can't be tested with the above commands: the `fontenc` package pretends that it was never loaded to allow for repeated reloading with different options (see `ltoutenc.dtx` for details).

42.3.4 Declaring new options

Options for classes and packages are built using the same macros.

??→P.??`{`⟨*name*⟩`}{`⟨*code*⟩`}`. To define a builtin option, use

??→P.?? 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*`.

[1] v1.1i 2013/07/07 Correctly describe how the date in `\@ifpackagelater` is used

Possible uses for `\DeclareOption*` include:

??[→P. ??]⟨⟩ Do nothing. Silently accept unknown options. (This suppresses the usual warnings.)

`\DeclareOption*{\@unkownoptionerror}`
Complain about unknown local options. (The initial setting for package files.)

`\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{`⟨*pkg-name*⟩`}}`
Handle the the current option by passing it on to the package ⟨*pkg-name*⟩, which will presumably be loaded via `\RequirePackage` later in the file. This is useful for building 'extension' packages, that perhaps handle a couple of new options, but then pass everything else on to an existing package.

```
\DeclareOption*{\InputIfFileExists{xx-
    \CurrentOption.yyy}%
                    {}%
                    {\OptionNotUsed}}
```

Handle the option foo by loading the file `xx-foo.yyy` if it exists, otherwise do nothing, but declare that the option was not used. Actually the **??**[→P. ??] declaration is only needed if this is being used in class files, but does no harm in package files.

42.3.5 Safe Input Macros

??[→P. ??]`{`⟨*file*⟩`}{`⟨*then*⟩`}{`⟨*else*⟩`}`
Inputs ⟨*file*⟩ if it exists. Immediately before the input, ⟨*then*⟩ is executed. Otherwise ⟨*else*⟩ is executed. Note that these macros are not defined in this file, but rather in the `ltfiles` module (see Chapter **??**).

??[→P. ??] is similar to the above, but does not input the file.

One thing you might like to put in the ⟨*else*⟩ clause is **??**[→P. ??] This starts an interactive request for a filename, supplying default extensions. Just hitting return causes the whole input to be skipped and entering `x` quits the current run,

??[→P. ??] This has been redefined from the LaTeX2.09 definition, in terms of the new commands `\InputIfFileExists` and `\@missingfileerror`.

`\listfiles` Giving this declaration in the preamble causes a list of all files input via the 'safe input' commands to be listed at the end. Any strings specified in the optional argument to

`\ProvidesPackage` are listed alongside the file name. So files in standard (and other non-standard) distributions can put informative strings in this argument.

42.4 Implementation

```
482 %<*2ekernel>
\if@compatibility
```

```
\newif\if@compatibility
```

\@documentclasshook

```
\def\@documentclasshook{%
   \ifx\@normalsize\@undefined
      \let\@normalsize\normalsize
   \fi
}
```

\@declaredoptions

This list is automatically built by **\DeclareOption**. It is the list of options (separated by commas) declared in the class or package file and it defines the order in which the the corresponding **\ds@**⟨*option*⟩ commands are executed. All local ⟨*option*⟩s which are not declared will be processed in the order defined by the optional argument of **\documentclass** or **\usepackage**.

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

\@classoptionslist

```
\let\@classoptionslist\relax
\@onlypreamble\@classoptionslist
```

\@unusedoptionlist

```
\let\@unusedoptionlist\@empty
\@onlypreamble\@unusedoptionlist
```

\CurrentOption

Name of current package or option.

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

\@currname

Name of current package or option.

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

\@currext

```
\global\let\@currext=\@empty
```

\@clsextension

\@pkgextension

```
\def\@clsextension{cls}
\def\@pkgextension{sty}
\@onlypreamble\@clsextension
\@onlypreamble\@pkgextension
```

\@pushfilename

\@popfilename

\@currnamestack

Commands to push and pop the file name and extension.
#1 current name.
#2 current extension.
#3 current catcode of @.
#4 Rest of the stack.

```
\def\@pushfilename{%
  \xdef\@currnamestack{%
    {\@currname}%
    {\@currext}%
    {\the\catcode`\@}%
    \@currnamestack}}
\@onlypreamble\@pushfilename
```

```
\def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil}
\@onlypreamble\@popfilename
```

```
\def\@p@pfilename#1#2#3#4\@nil{%
  \gdef\@currname{#1}%
  \gdef\@currext{#2}%
  \catcode`\@#3\relax
  \gdef\@currnamestack{#4}}
\@onlypreamble\@p@pfilename
```

```
\gdef\@currnamestack{}
\@onlypreamble\@currnamestack
```

\@ptionlist

```
\def\@ptionlist#1{%
  \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}}
\@onlypreamble\@ptionlist
```

The following two commands check to see if a class or a package has been loaded. The test is done by checking that `ver@pkgname.sty` or `ver@classname.cls` are defined. The test is carried comparing to `\relax`.

\@ifpackageloaded ⟨*package name*⟩⟨*true code*⟩⟨*false code*⟩

\@ifclassloaded ⟨*class name*⟩⟨*true code*⟩⟨*false code*⟩

`\@ifpackageloaded{`⟨*name*⟩`}` Checks to see whether a file has been loaded.

```
\def\@ifpackageloaded{\@ifl@aded\@pkgextension}
\def\@ifclassloaded{\@ifl@aded\@clsextension}
\@onlypreamble\@ifpackageloaded
\@onlypreamble\@ifclassloaded
```

```
\def\@ifl@aded#1#2{%
  \expandafter\ifx\csname ver@#2.#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}
\@onlypreamble\@ifl@aded
```

\@ifpackagelater

\@ifclasslater

```
\def\@ifpackagelater{\@ifl@ter\@pkgextension}
\def\@ifclasslater{\@ifl@ter\@clsextension}
\@onlypreamble\@ifpackagelater
\@onlypreamble\@ifclasslater
```

```
\def\@ifl@ter#1#2{%
  \expandafter\@ifl@t@r
    \csname ver@#2.#1\endcsname}
\@onlypreamble\@ifl@ter
```

```
\def\@ifl@t@r#1#2{%
  \ifnum\expandafter\@parse@version#1//00\@nil<%
        \expandafter\@parse@version#2//00\@nil
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}
\@onlypreamble\@ifl@t@r
```

```
\def\@parse@version#1/#2/#3#4#5\@nil{#1#2#3#4 }
\@onlypreamble\@parse@version
```

\@ifpackagewith

\@ifclasswith

```
\def\@ifpackagewith{\@if@ptions\@pkgextension}
\def\@ifclasswith{\@if@ptions\@clsextension}
\@onlypreamble\@ifpackagewith
\@onlypreamble\@ifclasswith
```

```
\def\@if@ptions#1#2{%
  \@expandtwoargs\@if@pti@ns{\@ptionlist{#2.#1}}}
\@onlypreamble\@if@ptions
```

```
\def\@if@pti@ns#1#2{%
 \let\reserved@a\@firstoftwo
 \@for\reserved@b:=#2\do{%
   \ifx\reserved@b\@empty
   \else
     \expandafter\in@\expandafter{\expandafter,\reserved@b,}{,#1,}%
     \ifin@
     \else
       \let\reserved@a\@secondoftwo
     \fi
   \fi
 }%
 \reserved@a}
\@onlypreamble\@if@pti@ns
```

\ProvidesPackage

Checks that the current filename is correct, and defines `\ver@filename`.

```
\def\ProvidesPackage#1{%
  \xdef\@gtempa{#1}%
  \ifx\@gtempa\@currname\else
    \@latex@warning@no@line{You have requested
      \@cls@pkg\space`\@currname',\MessageBreak
       but the \@cls@pkg\space provides `#1'}%
  \fi
  \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}%]
\@onlypreamble\ProvidesPackage
```

```
\def\@pr@videpackage[#1]{%
  \expandafter\xdef\csname ↊
       ↋ver@\@currname.\@currext\endcsname{#1}%
  \ifx\@currext\@clsextension
    \typeout{Document Class: \@gtempa\space#1}%
  \else
    \wlog{Package: \@gtempa\space#1}%
  \fi}
\@onlypreamble\@pr@videpackage
```

\ProvidesClass

```
\let\ProvidesClass\ProvidesPackage
\@onlypreamble\ProvidesClass
```

\ProvidesFile

\@providesfile

```
\def\ProvidesFile#1{%
  \begingroup
    \catcode`\ 10 %
    \ifnum \endlinechar<256 %
      \ifnum \endlinechar>\m@ne
        \catcode\endlinechar 10 %
      \fi
    \fi
    \@makeother\/%
    \@makeother\&%
    \kernel@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}}
```

42.5 Passing Options

With the basic commands behind us, the passing of options to a package or class is done through two commands `\PassOptionsToPackage` and `\PassOptionsToClass`.

\PassOptionsToPackage

\PassOptionsToClass

If the package has been loaded, we check that it was first loaded with the options. Otherwise we add the option list to that of the package.

```
\def\@pass@ptions#1#2#3{%
  \expandafter\xdef\csname opt@#3.#1\endcsname{%
    \@ifundefined{opt@#3.#1}\@empty
      {\csname opt@#3.#1\endcsname,}%
    \zap@space#2 \@empty}}
\@onlypreamble\@pass@ptions
```

```
\def\PassOptionsToPackage{\@pass@ptions\@pkgextension}
\def\PassOptionsToClass{\@pass@ptions\@clsextension}
\@onlypreamble\PassOptionsToPackage
\@onlypreamble\PassOptionsToClass
```

42.6 Declaring Options

Options are declared using the **DeclareOption** command.

\DeclareOption

\DeclareOption*

Adds an option as a **\ds@** command, or the default **\default@ds** command.

```
\def\DeclareOption{%
  \let\@fileswith@pti@ns\@badrequireerror
  \@ifstar\@defdefault@ds\@declareoption}

\long\def\@declareoption#1#2{%
   \xdef\@declaredoptions{\@declaredoptions,#1}%
   \toks@{#2}%
   \expandafter\edef\csname ds@#1\endcsname{\the\toks@}}

\long\def\@defdefault@ds#1{%
  \toks@{#1}%
  \edef\default@ds{\the\toks@}}

\@onlypreamble\DeclareOption
\@onlypreamble\@declareoption
\@onlypreamble\@defdefault@ds
```

\OptionNotUsed

```
\def\OptionNotUsed{%
  \ifx\@currext\@clsextension
    \xdef\@unusedoptionlist{%
      \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
      \CurrentOption}%
  \fi}
\@onlypreamble\OptionNotUsed
```

\default@ds

```
% \let\default@ds\OptionNotUsed
```

\ProcessOptions

\ProcessOptions*

\ProcessOptions calls **\ds@option** for each known package option, then calls **\default@ds** for each option on the local options list. Finally resets all the declared options to **\relax**. The empty option does nothing, this has to be reset on the off chance it's set to **\relax** if an empty element gets into the **\@declaredoptions** list.

The star form is similar but executes options given in the order specified in the document, not the order they are declared in the file. In the case of packages, global options are executed before local ones.

```
\def\ProcessOptions{%
  \let\ds@\@empty
  \edef\@curroptions{\@ptionlist{\@currname.\@currext}}%
  \@ifstar\@xprocess@ptions\@process@ptions}
\@onlypreamble\ProcessOptions
```

```
\def\@process@ptions{%
  \@for\CurrentOption:=\@declaredoptions\do{%
    \ifx\CurrentOption\@empty\else
      \@expandtwoargs\in@{,\CurrentOption,}{%
         ,\ifx\@currext\@clsextension\else\@classoptionslist,\fi
         \@curroptions,}%
      \ifin@
        \@use@ption
        \expandafter\let\csname ↊
              ↋ds@\CurrentOption\endcsname\@empty
      \fi
    \fi}%
  \@process@pti@ns}
\@onlypreamble\@process@ptions
```

```
\def\@xprocess@ptions{%
  \ifx\@currext\@clsextension\else
    \@for\CurrentOption:=\@classoptionslist\do{%
      \ifx\CurrentOption\@empty\else
        \@expandtwoargs\in@{,\CurrentOption,}{,\@declaredoptions,}%
        \ifin@
          \@use@ption
          \expandafter\let\csname ↊
                ↋ds@\CurrentOption\endcsname\@empty
        \fi
      \fi}%
  \fi
  \@process@pti@ns}
\@onlypreamble\@xprocess@ptions
```

The common part of **\ProcessOptions** and **\ProcessOptions***.

```
\def\@process@pti@ns{%
  \@for\CurrentOption:=\@curroptions\do{%
    \@ifundefined{ds@\CurrentOption}%
      {\@use@ption
       \default@ds}%
```

There should not be any non-empty definition of `\CurrentOption` at this point, as all the declared options were executed earlier. This is for compatibility with 2.09 styles which use `\def\ds@`... directly, and so have options which do not appear in `\@declaredoptions`.

```
      \@use@ption}%
```

Clear all the definitions for option code. First set all the declared options to `\relax`, then reset the 'default' and 'empty' options. and the lst of declared options.

```
  \@for\CurrentOption:=\@declaredoptions\do{%
    \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%
  \let\CurrentOption\@empty
  \let\@fileswith@pti@ns\@@fileswith@pti@ns
  \AtEndOfPackage{\let\@unprocessedoptions\relax}}
\@onlypreamble\@process@pti@ns
```

\@options

```
\def\@options{\ProcessOptions*}
\@onlypreamble\@options
```

\@use@ption

```
\def\@use@ption{%
  \@expandtwoargs\@removeelement\CurrentOption
  \@unusedoptionlist\@unusedoptionlist
  \csname ds@\CurrentOption\endcsname}
\@onlypreamble\@use@ption
```

\ExecuteOptions

```
\def\ExecuteOptions#1{%
  \def\reserved@a##1\@nil{%
    \@for\CurrentOption:=#1\do{\csname ↩
         ↪ds@\CurrentOption\endcsname}%
    \edef\CurrentOption{##1}}%
  \expandafter\reserved@a\CurrentOption\@nil}
\@onlypreamble\ExecuteOptions
```

The top-level commands, which just set some parameters then call the internal command, `\@fileswithoptions`.

\documentclass

The main new-style class declaration.

```
\def\documentclass{%
  \let\documentclass\@twoclasseserror
  \if@compatibility\else\let\usepackage\RequirePackage\fi
  \@fileswithoptions\@clsextension}
\@onlypreamble\documentclass
```

\documentstyle

```
\def\documentstyle{%
  \makeatletter\input{latex209.def}\makeatother
  \documentclass}
\@onlypreamble\documentstyle
```

\RequirePackage

```
\def\RequirePackage{%
  \@fileswithoptions\@pkgextension}
\@onlypreamble\RequirePackage
```

\LoadClass

```
\def\LoadClass{%
  \ifx\@currext\@pkgextension
     \@latex@error
      {\noexpand\LoadClass in package file}%
      {You may only use \noexpand\LoadClass in a class file.}%
  \fi
  \@fileswithoptions\@clsextension}
\@onlypreamble\LoadClass
```

\@loadwithoptions

```
\def\@loadwithoptions#1#2#3{%
  \expandafter\let\csname opt@#3.#1\expandafter\endcsname
       \csname opt@\@currname.\@currext\endcsname
   #2{#3}}
\@onlypreamble\@loadwithoptions
```

\LoadClassWithOptions

```
\def\LoadClassWithOptions{%
  \@loadwithoptions\@clsextension\LoadClass}
\@onlypreamble\LoadClassWithOptions
```

\RequirePackageWithOptions

```
\def\RequirePackageWithOptions{%
  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
  \@loadwithoptions\@pkgextension\RequirePackage}
\@onlypreamble\RequirePackageWithOptions
```

\usepackage

```
\def\usepackage#1#{%
  \@latex@error
    {\noexpand \usepackage before \string\documentclass}%
    {\noexpand \usepackage may only appear in the document
      preamble, i.e.,\MessageBreak
      between \noexpand\documentclass and
      \string\begin{document}.}%
  \@gobble}
\@onlypreamble\usepackage
```

\NeedsTeXFormat {⟨*format name*⟩}[⟨*version*⟩]

Check that the document is running on the correct system. The command checks first that the format that is proceesing the file is the same as that requested by the argument to **\NeedsTeXFormat** and if it passes it calls **??** → P. ?? to process the argument in square brackets. [2] [3] [4]

```
\def\NeedsTeXFormat#1{%
  \def\reserved@a{#1}%
  \ifx\reserved@a\fmtname
    \expandafter\@needsformat
  \else
     \@latex@error{This file needs format `\reserved@a'%
       \MessageBreak but this is `\fmtname'}{%
       The current input file will not be processed
       further,\MessageBreak
       because it was written for some other flavor of
       TeX.\MessageBreak\@ehd}%
```

If the file is not meant to be processed by LaTeX 2_ε inputting is stopped, without ending the run. It just ends inputting the current file.[5]

```
     \endinput \fi}
\@onlypreamble\NeedsTeXFormat
```

\@needsformat

As described earlier **\@needsformat** process the '[]' by first checking it is available and then calling **\@needsf@rmat**. The latter compares the release and issues a warning if the version is not available and an earlier one is being used.

```
\def\@needsformat{%
  \@ifnextchar[%]
    \@needsf@rmat
    {}}
\@onlypreamble\@needsformat
```

```
\def\@needsf@rmat[#1]{%
    \@ifl@t@r\fmtversion{#1}{}%
    {\@latex@warning@no@line
        {You have requested release `#1' of LaTeX,\MessageBreak
         but only release `\fmtversion' is available}}}
\@onlypreamble\@needsf@rmat
```

\zap@space

\zap@space foo⟨*space*⟩**\@empty** removes all spaces from **foo** that are not protected by { } groups.

```
\def\zap@space#1 #2{%
  #1%
  \ifx#2\@empty\else\expandafter\zap@space\fi
  #2}
```

[2] v0.2a 1993/11/14 made more robust for alternative syntax for other formats.
[3] v0.2c 1993/11/17 Name changed from **\NeedsFormat**
[4] v0.2d 1993/11/18 **\fmtname** **\fmtversion** not **\@**...
[5] v1.0h 1994/05/23 Don't stop completely when format is wrong

\@fileswithoptions

The common part of **\documentclass** and **\usepackage**. This is the most important part of the code, since when you say **\documentclass** it will start processing the document.

```
\def\@fileswithoptions#1{%
  \@ifnextchar[%]
    {\@fileswith@ptions#1}%
    {\@fileswith@ptions#1[]}}
\@onlypreamble\@fileswithoptions
```

Chages [6] [7]

```
\def\@fileswith@ptions#1[#2]#3{%
  \@ifnextchar[%]
  {\@fileswith@pti@ns#1[{#2}]#3}%
  {\@fileswith@pti@ns#1[{#2}]#3[]}}
\@onlypreamble\@fileswith@ptions
```

Then we do some work.

First of all, we define the global variables. Then we look to see if the file has already been loaded. If it has, we check that it was first loaded with at least the current options. If it has not, we add the current options to the package options, set the default version to be **0000/00/00**, and load the file if we can find it. Then we check the version number.

Finally, we restore the old file name, reset the default option, and we set the catcode of **@**.

For classes, we can immediately process the file. For other types, **#2** could be a comma separated list, so loop through, processing each one separately. [8] [9] [10] [11] [12]

```
\def\@fileswith@pti@ns#1[#2]#3[#4]{%
  \ifx#1\@clsextension
    \ifx\@classoptionslist\relax
      \xdef\@classoptionslist{\zap@space#2 \@empty}%
      \def\reserved@a{%
        \@onefilewithoptions#3[{#2}][{#4}]#1%
        \@documentclasshook}%
    \else
      \def\reserved@a{%
        \@onefilewithoptions#3[{#2}][{#4}]#1}%
    \fi
  \else
```

build up a list of calls to **\@onefilewithoptions** (one for each package) without thrashing the parameter stack.

```
    \def\reserved@b##1,{%
      \ifx\@nil##1\relax\else
        \ifx\relax##1\relax\else
         \noexpand\@onefilewithoptions##1[{#2}][{#4}]%
```

[6] v0.2f 1993/11/22 Made the default [] not [**\@unknownversion**]
[7] v1.1h 2007/08/05 Prevent loss of brackets PR/3965
[8] v0.2q 1993/12/17 Add **\@compatibility** hook
[9] v0.2s 1994/01/17 Modify to reduce parameter stack usage
[10] v0.2y 1994/02/07 Run **\@compatibility** on the first class to start (not the first to finish)
[11] v0.2z 1994/02/10 Renamed **\@compatibility** to **\@documentclasshook**. ASAJ.
[12] v1.1h 2007/08/05 Prevent loss of brackets PR/3965

```
          \noexpand\@pkgextension
        \fi
        \expandafter\reserved@b
      \fi}%
      \edef\reserved@a{\zap@space#3 \@empty}%
      \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nil,}%
  \fi
  \reserved@a}
\@onlypreamble\@fileswith@pti@ns
```

Have the main argument as `#1`, so we only need one `\expandafter` above. [13] [14] [15]

```
\def\@onefilewithoptions#1[#2][#3]#4{%
  \@pushfilename
  \xdef\@currname{#1}%
  \global\let\@currext#4%
  \expandafter\let\csname\@currname.\@currext-h@@k\endcsname\@empty
  \let\CurrentOption\@empty
  \@reset@ptions
  \makeatletter
```

Grab everything in a macro, so the parameter stack is popped before any processing begins. [16] [17]

```
  \def\reserved@a{%
    \@ifl@aded\@currext{#1}%
      {\@if@ptions\@currext{#1}{#2}{}%
        {\@latex@error
            {Option clash for \@cls@pkg\space #1}%
            {The package #1 has already been loaded
             with options:\MessageBreak
             \space\space[\@ptionlist{#1.\@currext}]\MessageBreak
             There has now been an attempt to load it
              with options\MessageBreak
             \space\space[#2]\MessageBreak
             Adding the global options:\MessageBreak
             \space\space
                  \@ptionlist{#1.\@currext},#2\MessageBreak
             to your \noexpand\documentclass declaration may ↱
                  ↳fix this.%
             \MessageBreak
             Try typing \space <return> \space to proceed.}}}%
      {\@pass@ptions\@currext{#2}{#1}%
       \global\expandafter
       \let\csname ver@\@currname.\@currext\endcsname\@empty
       \InputIfFileExists
         {\@currname.\@currext}%
         {}%
         {\@missingfileerror\@currname\@currext}%
```

[13] v0.2a 1993/11/14 Moved resetting of `\default@ds`, `\ds@` and `\@declaredoptions` here, from the end of `\ProcessOptions`.

[14] v0.2f 1993/11/22 Made the initial version [] not [`\@unknownversion`]

[15] v0.2m 1993/12/07 Reset `\CurrentOption`

[16] v0.2s 1994/01/17 Modify to reduce parameter stack usage

[17] v1.1b 1998/05/07 Modify help message for latex/2805

`\@unprocessedoptions` will generate an error for each specified option in a package unless a `\ProcessOptions` has appeared in the package file.

```
    \let\@unprocessedoptions\@@unprocessedoptions
    \csname\@currname.\@currext-h@@k\endcsname
    \expandafter\let\csname\@currname.\@currext-h@@k\endcsname
              \@undefined
    \@unprocessedoptions}
```

```
    \@ifl@ter\@currext{#1}{#3}{}%
      {\@latex@warning@no@line
         {You have requested,\on@line,
          version\MessageBreak
            `#3' of \@cls@pkg\space #1,\MessageBreak
          but only version\MessageBreak
           `\csname ver@#1.\@currext\endcsname'\MessageBreak
          is available}}%
```

```
    \ifx\@currext\@clsextension\let\LoadClass\@twoloadclasserror\fi
    \@popfilename
    \@reset@ptions}%
  \reserved@a}
\@onlypreamble\@onefilewithoptions
```

`\@@fileswith@pti@ns`

Save the definition (for error checking). [18]

```
\let\@@fileswith@pti@ns\@fileswith@pti@ns
\@onlypreamble\@@fileswith@pti@ns
```

`\@reset@ptions`

Reset the default option, and clear lists of declared options.

```
\def\@reset@ptions{%
  \global\ifx\@currext\@clsextension
    \let\default@ds\OptionNotUsed
   \else
    \let\default@ds\@unknownoptionerror
  \fi
  \global\let\ds@\@empty
  \global\let\@declaredoptions\@empty}
\@onlypreamble\@reset@ptions
```

42.6.1 Hooks

Allow code do be saved to be executed at specific later times.

Save things in macros, I considered using toks registers, (and `\addto@hook` from the NFSS code, that would require stacking the contents in the case of required packages, so just generate a new macro for each package.

`\@begindocumenthook`

[18] v0.2c 1993/11/17 Macro added

\@enddocumenthook

Stuff to appear at the beginning or end of the document.

```
\ifx\@begindocumenthook\@undefined
  \let\@begindocumenthook\@empty
\fi
\let\@enddocumenthook\@empty
```

\g@addto@macro

Globally add to the end of a macro.

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

\AtEndOfPackage

\AtEndOfClass

\AtBeginDocument

\AtEndDocument

```
\def\AtEndOfPackage{%
  \expandafter\g@addto@macro\csname\@currname.\@currext-h@@k\endcsname}
\let\AtEndOfClass\AtEndOfPackage
\@onlypreamble\AtEndOfPackage
\@onlypreamble\AtEndOfClass
```

```
\def\AtBeginDocument{\g@addto@macro\@begindocumenthook}
\def\AtEndDocument{\g@addto@macro\@enddocumenthook}
\@onlypreamble\AtBeginDocument
```

\@cls@pkg

The current file type. Name changed to avoid clash with output routine.

```
\def\@cls@pkg{%
  \ifx\@currext\@clsextension
    document class%
  \else
    package%
  \fi}
\@onlypreamble\@cls@pkg
```

\@unknownoptionerror

Bad option.

```
\def\@unknownoptionerror{%
  \@latex@error
    {Unknown option `\CurrentOption' for ↱
          ↳\@cls@pkg\space`\@currname'}%
    {The option `\CurrentOption' was not declared in
     \@cls@pkg\space`\@currname', perhaps you\MessageBreak
```

```
      misspelled its name.
     Try typing \space <return>
     \space to proceed.}}
\@onlypreamble\@unknownoptionerror
```

\@@unprocessedoptions

```
\def\@@unprocessedoptions{%
  \ifx\@currext\@pkgextension
    \edef\@curroptions{\@ptionlist{\@currname.\@currext}}%
    \@for\CurrentOption:=\@curroptions\do{%
        \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
  \fi}
\@onlypreamble\@unprocessedoptions
\@onlypreamble\@@unprocessedoptions
```

\@badrequireerror

```
\def\@badrequireerror#1[#2]#3[#4]{%
  \@latex@error
    {\noexpand\RequirePackage or \noexpand\LoadClass
         in Options Section}%
    {The \@cls@pkg\space `\@currname' is ↲
         ↳defective.\MessageBreak
     It attempts to load `#3' in the options section, ↲
          ↳i.e.,\MessageBreak
     between \noexpand\DeclareOption and ↲
          ↳\string\ProcessOptions.}}
\@onlypreamble\@badrequireerror
```

\@twoloadclasserror

Two `\LoadClass` in a class.

```
\def\@twoloadclasserror{%
  \@latex@error
    {Two \noexpand\LoadClass commands}%
    {You may only use one \noexpand\LoadClass in a class file}}
\@onlypreamble\@twoloadclasserror
```

\@twoclasseserror

Two `\documentclass` or `\documentstyle`. [19]

```
\def\@twoclasseserror#1#{%
  \@latex@error
    {Two \noexpand\documentclass or \noexpand\documentstyle ↲
         ↳commands}%
    {The document may only declare one class.}\@gobble}
\@onlypreamble\@twoclasseserror
```

[19] v0.2h 1993/11/28 Macro added

42.6.2 Providing shipment

\two@digits

Prefix a number less than 10 with '0'.

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

\filecontents

\endfilecontents

This environment implements inline files. The star-form does not write extra comments into the file. [20] [21] [22] [23] [24] [25]

```
\begingroup%
\catcode`\*=11 %
\catcode`\^^M\active%
\catcode`\^^L\active\let^^L\relax%
\catcode`\^^I\active%
\gdef\filecontents{\@tempswatrue\filec@ntents}%
\gdef\filecontents*{\@tempswafalse\filec@ntents}%
\gdef\filec@ntents#1{%
  \openin\@inputcheck#1 %
  \ifeof\@inputcheck%
    \@latex@warning@no@line%
        {Writing file `\@currdir#1'}%
    \chardef\reserved@c15 %
    \ch@ck7\reserved@c\write%
    \immediate\openout\reserved@c#1\relax%
  \else%
    \closein\@inputcheck%
    \@latex@warning@no@line%
            {File `#1' already exists on the ⤸
                 ⤹system.\MessageBreak%
             Not generating it from this source}%
    \let\write\@gobbletwo%
    \let\closeout\@gobble%
  \fi%
  \if@tempswa%
```

[26]

```
    \immediate\write\reserved@c{%
      \@percentchar\@percentchar\space%
          \expandafter\@gobble\string\LaTeX2e file `#1'^^J%
      \@percentchar\@percentchar\space  generated by the %
        `\@currenvir' ⤸
             ⤹\expandafter\@gobblefour\string\newenvironment^^J%
```

[20] v0.2h 1993/11/28 Don't globally allocate a write stream (always use 15)
[21] v0.2r 1993/12/19 Different message when ignoring a file
[22] v0.3g 1994/04/11 Add star form, dont write `\endinput` at the end of the file.
[23] v1.0c 1994/05/11 Add checks for form feed and tab
[24] v1.0m 1995/04/21 Close input check stream: latex/1487
[25] v1.0p 1995/05/25 Delete `\filec@ntents` after preamble
[26] v1.0y 1997/10/10 `\@currenvir` in banner

```
      \@percentchar\@percentchar\space from source ↩
           ↪`\jobname' on %
         \number\year/\two@digits\month/\two@digits\day.^^J%
      \@percentchar\@percentchar}%
  \fi%
  \let\do\@makeother\dospecials%
```

[27]

```
  \edef\E{\@backslashchar end\string{\@currenvir\string}}%
  \edef\reserved@b{%
    \def\noexpand\reserved@b%
        ####1\E####2\E####3\relax}%
  \reserved@b{%
    \ifx\relax##3\relax%
```

There was no `\end{filecontents}`

```
      \immediate\write\reserved@c{##1}%
    \else%
```

There was a `\end{filecontents}`, so stop this time.

```
      \edef^^M{\noexpand\end{\@currenvir}}%
      \ifx\relax##1\relax%
      \else%
```

Text before the `\end`, write it with a warning.

```
          \@latex@warning{Writing text `##1' before %
             \string\end{\@currenvir}\MessageBreak as last ↩
                  ↪line of #1}%
        \immediate\write\reserved@c{##1}%
      \fi%
      \ifx\relax##2\relax%
      \else%
```

Text after the `\end`, ignore it with a warning.

```
         \@latex@warning{%
           Ignoring text `##2' after \string\end{\@currenvir}}%
      \fi%
    \fi%
    ^^M}%

  \catcode`\^^L\active%
  \let\L\@undefined%
  \def^^L{\@ifundefined L^^J^^J^^J}%
  \catcode`\^^I\active%
  \let\I\@undefined%
  \def^^I{\@ifundefined I\space\space}%
  \catcode`\^^M\active%
  \edef^^M##1^^M{%
    \noexpand\reserved@b##1\E\E\relax}}%
\endgroup%

\begingroup
\catcode`|=\catcode`\%
\catcode`\%=12
\catcode`\*=11
```

[27] v1.0y 1997/10/10 Check for text before or after `\end` environment. latex/2636

```
\gdef\@percentchar{%}
\gdef\endfilecontents{|
  \immediate\closeout\reserved@c
  \def\T##1##2##3{|
  \ifx##1\@undefined\else
    \@latex@warning@no@line{##2 has been converted to Blank ↊##3e}|
  \fi}|
  \T\L{Form Feed}{Lin}|
  \T\I{Tab}{Spac}|
  \immediate\write\@unused{}}
\global\let\endfilecontents*\endfilecontents
\@onlypreamble\filecontents
\@onlypreamble\endfilecontents
\@onlypreamble\filecontents*
\@onlypreamble\endfilecontents*
\endgroup
\@onlypreamble\filec@ntents
```

```
%</2ekernel>
```

42.7 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```
%<*afterpreamble>
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{pkgindoc}
         [1994/10/20 v1.1 Package Interface in Document (DPC)]
\def\reserved@a#1\do\@classoptionslist#2\do\filec@ntents#3\relax{%
  \gdef\@preamblecmds{#1#3}}
\expandafter\reserved@a\@preamblecmds\relax
%</afterpreamble>
```

phd package and give a background as to how fonts are used in LaTeX.

This file contains the code for loading hyphenation patterns into LaTeX. Most of this will end up in a file called `hyphen.ltx`. If you wish to customize your LaTeX system in respect of hyphenation patterns, write a file `hyphen.cfg`. If this file exists, it will be loaded instead of `hyphen.ltx`. See the comments below for additional information.

```
%<*driver>
%\documentclass{ltxdoc}
%\begin{document}
%\DocInput{lthyphen.dtx}
%\end{document}
%</driver>
```

The default file **hyphen.ltx** loads hyphenation patterns for US english. If you want to load additional or other hyphenation patterns, you should create a file **hyphen.cfg**. This is best done by starting from **hyphen.ltx**.

For backward compatibility, the default file, **hyphen.ltx**, first tries to load the file **hyphen.tex**. If this file exists, an information message is issued and the appropriate defaults for TeX's internal parameters are set: **\language** is initialized to 0, and **\lefthyphenmin** and **\righthyphenmin** to 2 and 3, respectively, to disallow x- or -xx breaks.

```
%<*default>
\InputIfFileExists{hyphen.tex}%
   {\message{Loading hyphenation patterns for US ↩
        ↪english.}%
    \language=0
    \lefthyphenmin=2 \righthyphenmin=3 }%

   {\errhelp{The configuration for hyphenation is ↩
        ↪incorrectly
             installed.^^J%
             If you don't understand this error ↩
                  ↪message you need
             to seek^^Jexpert advice.}%
    \errmessage{OOPS! I can't find any hyphenation ↩
         ↪patterns for
                US english.^^J \space Think of ↩
                     ↪getting some or the
                latex2e setup will never ↩
                     ↪succeed}\@@end}
%</default>
```

The following example describes the possible contents of a file **hyphen.cfg** that will load both US English and German hyphenation patterns, making the former the default. It sets **\language** to 0 for the US patterns and to 1 for the German patterns. Then **\language** is set to 0 to make this the default and the default values of **\lefthyphenmin** and **\righthyphenmin** are set.

```
\language=0
\input hyphen % (or \input ushyphen1 if the file has been renamed)
\language=1
```

```
\input ghyph31
\language=0
\lefthyphenmin=2
\righthyphenmin=3
\endinput
```

Another possibility is to use the package **babel**, by Johannes Braams. That package is distributed with a suitable **hyphen.cfg** file.

phd package and give a background as to how fonts are used in LaTeX.

This section contains the final settings for LaTeX. It initialises some debugging and typesetting parameters, sets the default `\catcode`s and uc/lc codes, and inputs the hyphenation file.

The latest version also take care of some necessary XeLaTeX parameters.

44.0.1 Debugging

By default, LaTeX shows statistics:

```
%<*2ekernel>
\tracingstats1
```

44.0.2 Typesetting parameters

`\@lowpenalty` `\@medpenalty` `\@highpenalty`

These are penalties used internally.

```
\newcount\@lowpenalty
\newcount\@medpenalty
\newcount\@highpenalty
```

(*End 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.

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                    {\newmarks}{Extended ↩
      ↪Allocation}%
```

If eTeX is available, an allocation command is made available. `\marks` is an etex primitive command. It extends the **??**→P.?? command of the original tex. You can use it to store text on the current page which you want to use at shipout in the headers.

```
\ifx\marks\@undefined\else
\def\newmarks{%
  \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\e@alloc@top}
\fi
```

```
\documentclass{article}
\usepackage{etex}
\newmarks\mymark
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\lhead{\topmarks\mymark, \botmarks\mymark, \firstmarks\mymark}
\begin{document}
text
\marks\mymark{A first mark}

\marks\mymark{A second mark}

\newpage

```

```
text \marks\mymark{A third mark}
\end{document}
```

See TX.SX
macro:->\e@alloc \marks \e@alloc@chardef {\count 256}\m@ne \e@alloc@top

```
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\newmarks}{Extended Allocation}%
%<latexrelease>\let\newmarks\@undefined
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

(*End definition for* `\newmarks`.)

```
%</2ekernel>
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>                 {\newXeTeXintercharclass}{Extended Allocation}%
\ifx\XeTeXcharclass\@undefined
\else
\ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
  \chardef\e@alloc@intercharclass@top=4095
\else
  \chardef\e@alloc@intercharclass@top=255
\fi
\def\newXeTeXintercharclass{%
 \e@alloc\XeTeXcharclass
   \chardef\xe@alloc@intercharclass\m@ne\e@alloc@intercharclass@top}
\fi
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\newXeTeXintercharclass}{Extended Allocation}%
%<latexrelease> \ifx\XeTeXcharclass\@undefined
%<latexrelease> \else
%<latexrelease>    {\def\xe@alloc@#1#2#3#4#5{\global\advance#1\@ne
%<latexrelease>     \xe@ch@ck#1#4#2%
%<latexrelease>     \allocationnumber#1%
%<latexrelease>     \global#3#5\allocationnumber
%<latexrelease>     }{\wlog{\string#5=\string#2\the\allocationnumber}}
%<latexrelease>    \def\xe@ch@ck#1#2#3{%
%<latexrelease>     \ifnum#1<#2\else
%<latexrelease>      \errmessage{No room for a new #3}%
%<latexrelease>     \fi}
%<latexrelease>    \def\newXeTeXintercharclass{%
%<latexrelease>     \xe@alloc@\xe@alloc@intercharclass
```

```
%<latexrelease>                    ⤸
      ⤹\XeTeXcharclass\chardef\@cclv}
%<latexrelease> \fi
%<latexrelease>\EndIncludeInRelease
%<*2ekernel|latexrelease>
%<latexrelease>\IncludeInRelease{2016/02/01}%
%<latexrelease>  {\xe@alloc@intercharclass}{Start of XeTeX ⤸
      ⤹class allocator}%
\ifx\XeTeXcharclass\@undefined
\else
  \countdef\xe@alloc@intercharclass=257
  \xe@alloc@intercharclass=\z@
\fi
%</2ekernel|latexrelease>
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{2015/01/01}%
%<latexrelease>  {\xe@alloc@intercharclass}{Start of XeTeX ⤸
      ⤹class allocator}%
%<latexrelease> \ifx\XeTeXcharclass\@undefined
%<latexrelease> \else
%<latexrelease>   \xe@alloc@intercharclass=\thr@@
%<latexrelease> \fi
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>  {\xe@alloc@intercharclass}{Start of XeTeX ⤸
      ⤹class allocator}%
%<latexrelease> \ifx\XeTeXcharclass\@undefined
%<latexrelease> \else
%<latexrelease>   \newcount\xe@alloc@intercharclass
%<latexrelease>   \xe@alloc@intercharclass=\thr@@
%<latexrelease> \fi
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

```
\unitlength = 1pt
\fboxsep = 3pt
\fboxrule = .4pt
```

```
\@maxdepth       = \maxdepth
```

\vsize initialized because a **\clearpage** with **\vsize < \topskip** causes trouble. **\@colroom** and **\@colht** also initialized because **\vsize** may be set to them if a **\clearpage** is done before the **\begin{document}**

```
\vsize = 1000pt
\@colroom = \vsize
\@colht = \vsize
```

Initialise **\textheight** **\textwidth** and page style, to avoid internal errors if they are not set by the class.

```
\textheight=.5\maxdimen
\textwidth=\textheight
\ps@empty
```

44.0.3 Lccodes for hyphenation

For 7- and 8-bit engines the assumption of T1 encodings is the basis for the hyphenation patterns. That's not the case for the Unicode engines, where the assumption is engine-native working. The common loader system provides access to data from the Unicode Consortium covering not only `\lccode` but also other related data. The `\lccode` part of that at least needs to be loaded before hyphenation is tackled: XeTEX follows the standard TEX route of building patterns into the format. LuaTEX doesn't require this data be loaded *here* but it does need to be loaded somewhere. Rather than test for the Unicode engines by name, the approach here is to look for the extended math mode handling both provide: any other engine developed in this area will presumably also provide `\Umathcode`.

```
\ifnum 0%
  \ifx\Umathcode\@undefined\else 1\fi
  \ifx\XeTeXmathcode\@undefined\else 1\fi
  >\z@
  \message{ Unicode character data,}
  \input{load-unicode-data}
%</2ekernel>
%<latexrelease>\IncludeInRelease{2016/02/01}%
%<latexrelease>  {\XeTeXintercharclasses}{XeTeX character classes}%
%<latexrelease>  \ifx\XeTeXinterchartoks\undefined
%<latexrelease>  \else
%<latexrelease>    \begingroup
%<latexrelease>      \chardef\XeTeXcharclassID = 0 %
%<latexrelease>      \chardef\XeTeXcharclassOP = 0 %
%<latexrelease>      \chardef\XeTeXcharclassCL = 0 %
%<latexrelease>      \chardef\XeTeXcharclassEX = 0 %
%<latexrelease>      \chardef\XeTeXcharclassIS = 0 %
%<latexrelease>      \chardef\XeTeXcharclassNS = 0 %
%<latexrelease>      \chardef\XeTeXcharclassCM = 0 %
%<latexrelease>      \input{load-unicode-xetex-classes}
%<latexrelease>    \endgroup
%<latexrelease>    \global\let\xtxHanGlue\undefined
%<latexrelease>    \global\let\xtxHanSpace\undefined
%<latexrelease>    \global\XeTeXinterchartoks 0 1 = {}
%<latexrelease>    \global\XeTeXinterchartoks 0 2 = {}
%<latexrelease>    \global\XeTeXinterchartoks 0 3 = {}
%<latexrelease>    \global\XeTeXinterchartoks 1 0 = {}
%<latexrelease>    \global\XeTeXinterchartoks 2 0 = {}
%<latexrelease>    \global\XeTeXinterchartoks 3 0 = {}
%<latexrelease>    \global\XeTeXinterchartoks 1 1 = {}
%<latexrelease>    \global\XeTeXinterchartoks 1 2 = {}
%<latexrelease>    \global\XeTeXinterchartoks 1 3 = {}
%<latexrelease>    \global\XeTeXinterchartoks 2 1 = {}
%<latexrelease>    \global\XeTeXinterchartoks 2 2 = {}
%<latexrelease>    \global\XeTeXinterchartoks 2 3 = {}
%<latexrelease>    \global\XeTeXinterchartoks 3 1 = {}
%<latexrelease>    \global\XeTeXinterchartoks 3 2 = {}
%<latexrelease>    \global\XeTeXinterchartoks 3 3 = {}
```

```
%<latexrelease>  \fi
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>  {\XeTeXintercharclasses}{XeTeX character classes}%
%<latexrelease>  \ifx\XeTeXinterchartoks\undefined
%<latexrelease>  \else
%<latexrelease>   \input{load-unicode-xetex-classes}
%<latexrelease>   \gdef\xtxHanGlue{\hskip0pt plus 0.1em\relax}
%<latexrelease>   \gdef\xtxHanSpace{\hskip0.2em plus 0.2em minus 0.1em\relax}
%<latexrelease>   \global\XeTeXinterchartoks 0 1 = {\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 0 2 = {\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 0 3 = {\nobreak\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 1 0 = {\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 2 0 = {\nobreak\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 3 0 = {\xtxHanSpace}
%<latexrelease>   \global\XeTeXinterchartoks 1 1 = {\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 1 2 = {\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 1 3 = {\nobreak\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 2 1 = {\nobreak\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 2 2 = {\nobreak\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 2 3 = {\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 3 1 = {\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 3 2 = {\xtxHanGlue}
%<latexrelease>   \global\XeTeXinterchartoks 3 3 = {\nobreak\xtxHanGlue}
%<latexrelease>  \fi
%<latexrelease>\EndIncludeInRelease
%<*2ekernel>
```

```
  \lccode`\- =`\- % default hyphen char
```

```
\else
```

We set things up so that hyphenation files can assume that the default (T1) lc-codes are in use (at present this also sets up the uccodes). We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
\def\reserved@a#1#2{%
   \@tempcnta#1\relax
```

```
   \@tempcntb#2\relax
   \reserved@b
}
\def\reserved@b{%
   \ifnum\@tempcnta>\@tempcntb\else
      \reserved@c\@tempcnta
      \advance\@tempcnta\@ne
      \expandafter\reserved@b
   \fi
}
```

Depending on the TEX version, we might not be allowed to do this for non-ASCII characters.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by -"20
   \uccode#1=\count@
   \lccode#1=#1
}
\reserved@a{`\a}{`\z}
\reserved@a{"A0}{"BC}
\reserved@a{"E0}{"FF}
```

The upper case characters need their **\uccode** and **\lccode** values set, and their **\sfcode** set to 999.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by "20
   \uccode#1=#1
   \lccode#1=\count@
   \sfcode#1=999
}
\reserved@a{`\A}{`\Z}
\reserved@a{"80}{"9C}
\reserved@a{"C0}{"DF}
```

```
\uccode`\^^Y=`\I     % dotless i
\lccode`\^^Y=`\^^Y   % dotless i
\uccode`\^^Z=`\J     % dotless j, ae in OT1
\lccode`\^^Z=`\^^Z   % dotless j, ae in OT1
\lccode`\^^9d=`\i    % dotted I
\uccode`\^^9d=`\^^9d % dotted I
\lccode`\^^9e=`\^^9e % d-bar
\uccode`\^^9e=`\^^d0 % d-bar
```

```
\lccode`\^^[=`\^^[   % oe in OT1
```

And we also set the **\lccode** of **\-** and **\textcompwordmark** so that they do not prevent hyphenation in the remainder of the word (as suggested by Lars Helström).

```
\lccode`\- =`\-   % default hyphen char
\lccode 127=127   % alternate hyphen char
\lccode 23 =23    % textcompwordmark in T1
```

End of the conditional to select either Unicode or T1 encoding defaults.

```
\fi
```

This is as good a place as any to active a few XeTEX-specific settings

```
\ifx\XeTeXuseglyphmetrics\@undefined
```

```
\else
  \XeTeXuseglyphmetrics=1 %
  \XeTeXdashbreakstate=1 %
\fi
```

44.1 Hyphenation

The following code will be compiled into the format file. It checks for the existence of `hyphen.cfg` in inputs that file if found. Otherwise it inputs `hyphen.ltx`. Note that these are loaded in *before* the `\catcode`s are set, so local hyphenation files can use 8-bit input. We try to load the customized hyphenation description file.

```
\InputIfFileExists{hyphen.cfg}
           {\typeout{=============================================^^J%
                      Local configuration file hyphen.cfg ↊
                           ↊used^^J%
                     ==========================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
           }
           {\input{hyphen.ltx}}
\let\@addtofilelist\@gobble
```

```
\ifx\l@nohyphenation \@undefined
  \newlanguage\l@nohyphenation
\fi
```

`ent@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.

```
\let\document@default@language\m@ne
```

(*End definition for* `\document@default@language`.)

44.1.1 Font loading

Fonts loaded during the formatting process might already have changed the `\font@submax` from `0pt` to something higher. If so, we put out a bold warning.

```
\ifdim \font@submax >\z@
   \@font@warning{Size substitutions with ↊
        ↊differences\MessageBreak
                 up to \font@submax\space have ↊
                      ↊occurred.\MessageBreak
                \MessageBreak
                Please check the transcript file
                carefully\MessageBreak
                and redo the format generation if necessary!
                \@gobbletwo}%
   \errhelp{Only stopped, to give you time to
            read the above message.}
   \errmessage{}
```

We reset the macro. Otherwise every user will get a warning on every job.

```
\def\font@submax{0pt}
\fi
```

44.1.2 Input encoding

We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
\def\reserved@a#1#2{%
   \@tempcnta#1\relax
   \@tempcntb#2\relax
   \reserved@b
}
\def\reserved@b{%
   \ifnum\@tempcnta>\@tempcntb\else
      \reserved@c\@tempcnta
      \advance\@tempcnta\@ne
      \expandafter\reserved@b
   \fi
}
```

Set the special catcodes (although some of these are useless, since an error will have occurred if the catcodes have changed). Note that `^^J` has catcode 'other' for use in warning messages.

```
\catcode`\ =10
\catcode`\#=6
\catcode`\$=3
\catcode`\%=14
\catcode`\&=4
\catcode`\\=0
\catcode`\^=7
\catcode`\_=8
\catcode`\{=1
\catcode`\}=2
\catcode`\~=13
\catcode`\@=11
\catcode`\^^I=10
\catcode`\^^J=12
\catcode`\^^L=13
\catcode`\^^M=5
```

Set the 'other' catcodes.

```
\def\reserved@c#1{\catcode#1=12\relax}
\reserved@c{`\!}
\reserved@c{`\"}
\reserved@a{`\'}{`\?}
\reserved@c{`\[}
\reserved@c{`\]}
\reserved@c{`\`}
\reserved@c{`\|}
```

Set the 'letter' catcodes.

```
\def\reserved@c#1{\catcode#1=11\relax}
\reserved@a{`\A}{`\Z}
\reserved@a{`\a}{`\z}
```

All the characters in the range 0–31 and 127–255 are illegal, *except* tab (`^^I`), nl (`^^J`), ff (`^^L`) and cr (`^^M`). Now allow 8-bit characters, although their use in this way is strongly discouraged. See `inputenc.dtx` for a supported mechanism for 8-bit input.

```
\def\reserved@c#1{\catcode#1=15\relax}
\reserved@a{0}{`\^^H}
\reserved@c{`\^^K}
\reserved@a{`\^^N}{31}
```

44.1.3 Lccodes and uccodes

```
\ifnum 0%
  \ifx\Umathcode\@undefined\else 1\fi
  \ifx\XeTeXmathcode\@undefined\else 1\fi
  >\z@
\else
\def\reserved@c#1{%
   \count@=#1\advance\count@ by -"20
   \uccode#1=\count@
   \lccode#1=#1
}
\reserved@a{`\a}{`\z}
\reserved@a{"A0}{"BC}
\reserved@a{"E0}{"FF}
```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by "20
   \uccode#1=#1
   \lccode#1=\count@
   \sfcode#1=999
}
\reserved@a{`\A}{`\Z}
\reserved@a{"80}{"9C}
\reserved@a{"C0}{"DF}
```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose uccode or lccode isn't quite what you'd expect.

```
\uccode`\^^Y=`\I     % dotless i
\lccode`\^^Y=`\^^Y   % dotless i
\uccode`\^^Z=`\J     % dotless j, ae in OT1
\lccode`\^^Z=`\^^Z   % dotless j, ae in OT1
\lccode`\^^9d=`\i    % dotted I
\uccode`\^^9d=`\^^9d % dotted I
\lccode`\^^9e=`\^^9e % d-bar
\uccode`\^^9e=`\^^d0 % d-bar
```

```
\lccode`\^^[=`\^^[   % oe in OT1
\fi % End of reset block for 8-bit engines
```

```
\DeclareRobustCommand{\MakeUppercase}[1]{{%
      \def\i{I}\def\j{J}%
```

```
      \def\reserved@a##1##2{\let##1##2\reserved@a}%
      \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
      \protected@edef\reserved@a{\uppercase{#1}}%
      \reserved@a
   }}
\DeclareRobustCommand{\MakeLowercase}[1]{{%
      \def\reserved@a##1##2{\let##2##1\reserved@a}%
      \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
      \protected@edef\reserved@a{\lowercase{#1}}%
      \reserved@a
   }}
\def\@uclclist{\oe\OE\o\O\ae\AE
      \dh\DH\dj\DJ\l\L\ng\NG\ss\SS\th\TH}

\protected@edef\MakeUppercase#1{\MakeUppercase{#1}}
\protected@edef\MakeLowercase#1{\MakeLowercase{#1}}

%\IfFileExists{ltptch.ltx}
%  {\typeout{==================================^^J%
%             Applying patch file ltptch.ltx^^J%
%            ==================================}
%   \def\fmtversion@topatch{unknown}
%   \input{ltptch.ltx}
%   \ifx\fmtversion\fmtversion@topatch
%      \ifx\patch@level\@undefined
%        \typeout{^^J^^J^^J%
%         ↊
        ↊!!!
```

```
%    !! is not suitable for version <\fmtversion> of LaTeX.^^J^^J%
%    !! Please check if initex found an old patch file:^^J%
%    !! --- if so, rename it or delete it, and redo the^^J%
%    !!     initex run.^^J%
%    !!!!
```

phd package and give a background as to how fonts are used in LaTeX.

45.1 Overview

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

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

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

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

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

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

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

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

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

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

45.2 Core TEX functionality

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

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

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

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

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

\newluabytecode **\newluabytecode**{⟨*bytecode*⟩}
Allocates a number for Lua bytecode register, indexed from 1.

\newluachunkname **newluachunkname**{⟨*chunkname*⟩}
Allocates a number for Lua chunk register, indexed from 1. Also enters the

name of the regiser (without backslash) into the **lua.name** table to be used in stack traces.

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

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

\setattribute
\unsetattribute

\setattribute{⟨*attribute*⟩}{⟨*value*⟩}
\unsetattribute{⟨*attribute*⟩}

Set and unset attributes in a manner analogous to **\setlength**. Note that attributes take a marker value when unset so this operation is distinct from setting the value to zero.

45.3 Plain TEX interface

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

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

45.4 Lua functionality

45.4.1 Allocators in Lua

new_attribute

luatexbase.new_attribute(⟨*attribute*⟩**)**

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

new_whatsit

luatexbase.new_whatsit(⟨*whatsit*⟩**)**

Returns an allocation number for the custom ⟨*whatsit*⟩, indexed from 1.

new_bytecode

luatexbase.new_bytecode(⟨*bytecode*⟩**)**

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

new_chunkname

luatexbase.new_chunkname(⟨*chunkname*⟩**)**

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

new_luafunction

luatexbase.new_luafunction(⟨*functionname*⟩**)**

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

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

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

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

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

45.4.2 Lua access to TEX register numbers

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

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

As an example, consider the input:

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

\test{undefinedrubbish}

\test{space}

\test{hbox}

\test{@MM}

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

\test{strutbox}

\test{sixt@@n}

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

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

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

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

45.4.3 Module utilities

provides_module **luatexbase.provides_module(**⟨*info*⟩**)**

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

module_info **luatexbase.module_info(**⟨*module*⟩, ⟨*text*⟩**)**
module_warning **luatexbase.module_warning(**⟨*module*⟩, ⟨*text*⟩**)**
module_error **luatexbase.module_error(**⟨*module*⟩, ⟨*text*⟩**)**

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

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

45.4.4 Callback management

add_to_callback **luatexbase.add_to_callback(**⟨*callback*⟩, ⟨*function*⟩, ⟨*description*⟩**)**

Registers the ⟨*function*⟩ into the ⟨*callback*⟩ with a textual ⟨*description*⟩ of the function. Functions are inserted into the callback in the order loaded.

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

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

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

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

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

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

```
%<*2ekernel|tex|latexrelease>
%<2ekernel|latexrelease>\ifx\directlua\@undefined\else
%<latexrelease>\IncludeInRelease{2015/10/01}
%<latexrelease>                 {\newluafunction}{LuaTeX}%
\ifnum\luatexversion<60 %
  \wlog{***************************************************}
  \wlog{* LuaTeX version too old for ltluatex support *}
  \wlog{***************************************************}
  \expandafter\endinput
\fi
%<*tex>
\directlua{tex.enableprimitives("",tex.extraprimitives("luatex"))}
\ifx\e@alloc\@undefined
  \ifx\documentclass\@undefined
    \ifx\loccount\@undefined
      \input{etex.src}%
    \fi
    \catcode`\@=11 %
    \outer\expandafter\def\csname newfam\endcsname
                          {\alloc@8\fam\chardef\et@xmaxfam}
  \else
    \RequirePackage{etex}
    \expandafter\def\csname newfam\endcsname
                     {\alloc@8\fam\chardef\et@xmaxfam}
    \expandafter\let\expandafter\new@mathgroup\csname ↊
          ↋newfam\endcsname
  \fi
\edef \et@xmaxregs {\ifx\directlua\@undefined 32768\else ↊
        ↋65536\fi}
```

```
\edef \et@xmaxfam ⤸
      ⤹{\ifx\Umathchar\@undefined\sixt@@n\else\@cclvi\fi}
\count 270=\et@xmaxregs % locally allocates \count registers
\count 271=\et@xmaxregs % ditto for \dimen registers
\count 272=\et@xmaxregs % ditto for \skip registers
\count 273=\et@xmaxregs % ditto for \muskip registers
\count 274=\et@xmaxregs % ditto for \box registers
\count 275=\et@xmaxregs % ditto for \toks registers
\count 276=\et@xmaxregs % ditto for \marks classes
% \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}
\expandafter\let\csname ⤸
      ⤹newcount\expandafter\expandafter\endcsname
                 \csname globcount\endcsname
\expandafter\let\csname ⤸
      ⤹newdimen\expandafter\expandafter\endcsname
                 \csname globdimen\endcsname
\expandafter\let\csname ⤸
      ⤹newskip\expandafter\expandafter\endcsname
                 \csname globskip\endcsname
\expandafter\let\csname ⤸
      ⤹newbox\expandafter\expandafter\endcsname
                 \csname globbox\endcsname
\chardef\e@alloc@top=65535
\let\e@alloc@chardef\chardef
\def\e@alloc#1#2#3#4#5#6{%
  \global\advance#3\@ne
  \e@ch@ck{#3}{#4}{#5}#1%
  \allocationnumber#3\relax
  \global#2#6\allocationnumber
  \wlog{\string#6=\string#1\the\allocationnumber}}%
\gdef\e@ch@ck#1#2#3#4{%
  \ifnum#1<#2\else
    \ifnum#1=#2\relax
      #1\@cclvi
      \ifx\count#4\advance#1 10 \fi
    \fi
    \ifnum#1<#3\relax
    \else
      \errmessage{No room for a new \string#4}%
    \fi
  \fi}%
\long\def\@gobble#1{}
\long\def\@firstofone#1{#1}
\expandafter\csname newcount\endcsname\e@alloc@attribute@count
\expandafter\csname newcount\endcsname\e@alloc@ccodetable@count
\expandafter\csname ⤸
      ⤹newcount\endcsname\e@alloc@luafunction@count
\expandafter\csname newcount\endcsname\e@alloc@whatsit@count
\expandafter\csname newcount\endcsname\e@alloc@bytecode@count
\expandafter\csname newcount\endcsname\e@alloc@luachunk@count
```

```
\fi
%</tex>
```

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

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

```
\ifx\e@alloc@ccodetable@count\@undefined
  \countdef\e@alloc@ccodetable@count=259
\fi
\def\newcatcodetable#1{%
  \e@alloc\catcodetable\chardef
    \e@alloc@ccodetable@count\m@ne{"8000}#1%
  \initcatcodetable\allocationnumber
}
\e@alloc@ccodetable@count=\z@
```

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

```
    \parseunicodedataII#1;#3;#2 First>\relax
  }%
  \def\parseunicodedataII#1;#2;#3 First>#4\relax{%
    \ifx\relax#4\relax
      \expandafter\parseunicodedataIII
    \else
      \expandafter\parseunicodedataIV
    \fi
      {#1}#2\relax%
  }%
  \def\parseunicodedataIII#1#2#3\relax{%
    \ifnum 0%
      \if L#21\fi
      \if M#21\fi
      >0 %
      \catcode"#1=11 %
    \fi
  }%
  \def\parseunicodedataIV#1#2#3\relax{%
    \read\unicoderead to \unicodedataline
    \if L#2%
      \count0="#1 %
      \expandafter\parseunicodedataV\unicodedataline\relax
    \fi
  }%
  \def\parseunicodedataV#1;#2\relax{%
    \loop
      \unless\ifnum\count0>"#1 %
        \catcode\count0=11 %
        \advance\count0 by 1 %
    \repeat
  }%
  \def\storedpar{\par}%
  \chardef\unicoderead=\numexpr\count16 + 1\relax
  \openin\unicoderead=UnicodeData.txt %
  \loop\unless\ifeof\unicoderead %
    \read\unicoderead to \unicodedataline
    \unless\ifx\unicodedataline\storedpar
      \expandafter\parseunicodedataI\unicodedataline\relax
    \fi
  \repeat
  \closein\unicoderead
  \@firstofone{%
    \catcode64=12 %
    \savecatcodetable\catcodetable@latex
    \catcode64=11 %
    \savecatcodetable\catcodetable@atletter
   }
\endgroup
```

```
\ifx\e@alloc@luafunction@count\@undefined
  \countdef\e@alloc@luafunction@count=260
\fi
\def\newluafunction{%
  \e@alloc\luafunction\e@alloc@chardef
```

```
    \e@alloc@luafunction@count\m@ne\e@alloc@top
}
\e@alloc@luafunction@count=\z@
\ifx\e@alloc@whatsit@count\@undefined
  \countdef\e@alloc@whatsit@count=261
\fi
\def\newwhatsit#1{%
  \e@alloc\whatsit\e@alloc@chardef
    \e@alloc@whatsit@count\m@ne\e@alloc@top#1%
}
\e@alloc@whatsit@count=\z@
\ifx\e@alloc@bytecode@count\@undefined
  \countdef\e@alloc@bytecode@count=262
\fi
\def\newluabytecode#1{%
  \e@alloc\luabytecode\e@alloc@chardef
    \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
}
\e@alloc@bytecode@count=\z@
\ifx\e@alloc@luachunk@count\@undefined
  \countdef\e@alloc@luachunk@count=263
\fi
\def\newluachunkname#1{%
  \e@alloc\luachunk\e@alloc@chardef
    \e@alloc@luachunk@count\m@ne\e@alloc@top#1%
    {\escapechar\m@ne
    \directlua{lua.name[\the\allocationnumber]="\string#1"}}%
}
\e@alloc@luachunk@count=\z@
%<2ekernel>\everyjob\expandafter{%
%<2ekernel>  \the\everyjob
  \begingroup
    \attributedef\attributezero=0 %
    \chardef     \charzero     =0 %
    \countdef    \CountZero    =0 %
    \dimendef    \dimenzero    =0 %
    \mathchardef \mathcharzero =0 %
    \muskipdef   \muskipzero   =0 %
    \skipdef     \skipzero     =0 %
    \toksdef     \tokszero     =0 %
    \directlua{require("ltluatex")}
  \endgroup
%<2ekernel>}
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}
%<latexrelease>                  {\newluafunction}{LuaTeX}%
%<latexrelease>\let\e@alloc@attribute@count\@undefined
%<latexrelease>\let\newattribute\@undefined
%<latexrelease>\let\setattribute\@undefined
%<latexrelease>\let\unsetattribute\@undefined
%<latexrelease>\let\e@alloc@ccodetable@count\@undefined
```

```
%<latexrelease>\let\newcatcodetable\@undefined
%<latexrelease>\let\catcodetable@initex\@undefined
%<latexrelease>\let\catcodetable@string\@undefined
%<latexrelease>\let\catcodetable@latex\@undefined
%<latexrelease>\let\catcodetable@atletter\@undefined
%<latexrelease>\let\e@alloc@luafunction@count\@undefined
%<latexrelease>\let\newluafunction\@undefined
%<latexrelease>\let\e@alloc@luafunction@count\@undefined
%<latexrelease>\let\newwhatsit\@undefined
%<latexrelease>\let\e@alloc@whatsit@count\@undefined
%<latexrelease>\let\newluabytecode\@undefined
%<latexrelease>\let\e@alloc@bytecode@count\@undefined
%<latexrelease>\let\newluachunkname\@undefined
%<latexrelease>\let\e@alloc@luachunk@count\@undefined
%<latexrelease>\directlua{luatexbase.uninstall()}
%<latexrelease>\EndIncludeInRease
```

```
%<latexrelease>\IncludeInRelease{2017/01/01}%
%<latexrelease>                 {\fontencoding}{TU in ⤸
      ⤹everyjob}%
%<latexrelease>\fontencoding{TU}\let\encodingdefault\f@encoding
%<latexrelease>\ifx\directlua\@undefined\else
%<2ekernel>\everyjob\expandafter{%
%<2ekernel>  \the\everyjob
%<*2ekernel,latexrelease>
  \directlua{%
  if xpcall(function ()%
              require('luaotfload-main')%
            end,texio.write_nl) then %
  local _void = luaotfload.main ()%
  else %
  texio.write_nl('Error in luaotfload: reverting to OT1')%
  tex.print('\string\\def\string\\encodingdefault{OT1}')%
  end %
  }%
  \let\f@encoding\encodingdefault
  \expandafter\let\csname ⤸
        ⤹ver@luaotfload.sty\endcsname\fmtversion
%</2ekernel,latexrelease>
%<latexrelease>\fi
%<2ekernel>  }
%<latexrelease>\EndIncludeInRelease
%<latexrelease>\IncludeInRelease{0000/00/00}%
%<latexrelease>                 {\fontencoding}{TU in ⤸
      ⤹everyjob}%
%<latexrelease>\fontencoding{OT1}\let\encodingdefault\f@encoding
%<latexrelease>\EndIncludeInRelease
```

```
%<2ekernel|latexrelease>\fi
%</2ekernel|tex|latexrelease>
```

```
%<*lua>
```

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

```
local string_gsub      = string.gsub
```

```
local tex_count        = tex.count
local tex_setattribute = tex.setattribute
local tex_setcount     = tex.setcount
local texio_write_nl   = texio.write_nl
local luatexbase_warning
local luatexbase_error
local modules = modules or { }
local function luatexbase_log(text)
  texio_write_nl("log", text)
end
local function provides_module(info)
  if not (info and info.name) then
    luatexbase_error("Missing module name for provides_module")
  end
  local function spaced(text)
    return text and (" " .. text) or ""
  end
  luatexbase_log(
    "Lua module: " .. info.name
      .. spaced(info.date)
      .. spaced(info.version)
      .. spaced(info.description)
  )
  modules[info.name] = info
end
luatexbase.provides_module = provides_module
local function msg_format(mod, msg_type, text)
  local leader = ""
  local cont
  local first_head
  if mod == "LaTeX" then
    cont = string_gsub(leader, ".", " ")
    first_head = leader .. "LaTeX: "
  else
    first_head = leader .. "Module "  .. msg_type
    cont = "(" .. mod .. ")"
      .. string_gsub(first_head, ".", " ")
    first_head =  leader .. "Module "  .. mod .. " " .. msg_type  .. ":"
  end
  if msg_type == "Error" then
    first_head = "\n" .. first_head
  end
  if string.sub(text,-1) ~= "\n" then
    text = text .. " "
  end
  return first_head .. " "
    .. string_gsub(
         text
         .. "on input line "
         .. tex.inputlineno, "\n", "\n" .. cont .. " "
       )
```

```
  .. "\n"
end
local function module_info(mod, text)
  texio_write_nl("log", msg_format(mod, "Info", text))
end
luatexbase.module_info = module_info
local function module_warning(mod, text)
  texio_write_nl("term and log",msg_format(mod, "Warning", text))
end
luatexbase.module_warning = module_warning
local function module_error(mod, text)
  error(msg_format(mod, "Error", text))
end
luatexbase.module_error = module_error
function luatexbase_warning(text)
  module_warning("luatexbase", text)
end
function luatexbase_error(text)
  module_error("luatexbase", text)
end
local luaregisterbasetable = { }
local registermap = {
  attributezero = "assign_attr"    ,
  charzero      = "char_given"     ,
  CountZero     = "assign_int"     ,
  dimenzero     = "assign_dimen"   ,
  mathcharzero  = "math_given"     ,
  muskipzero    = "assign_mu_skip" ,
  skipzero      = "assign_skip"    ,
  tokszero      = "assign_toks"    ,
}
local createtoken
if tex.luatexversion > 81 then
  createtoken = token.create
elseif tex.luatexversion > 79 then
  createtoken = newtoken.create
end
local hashtokens    = tex.hashtokens()
local luatexversion = tex.luatexversion
for i,j in pairs (registermap) do
  if luatexversion < 80 then
    luaregisterbasetable[hashtokens[i][1]] =
      hashtokens[i][2]
  else
    luaregisterbasetable[j] = createtoken(i).mode
  end
end
local registernumber
if luatexversion < 80 then
  function registernumber(name)
    local nt = hashtokens[name]
```

```
    if(nt and luaregisterbasetable[nt[1]]) then
      return nt[2] - luaregisterbasetable[nt[1]]
    else
      return false
    end
  end
else
  function registernumber(name)
    local nt = createtoken(name)
    if(luaregisterbasetable[nt.cmdname]) then
      return nt.mode - luaregisterbasetable[nt.cmdname]
    else
      return false
    end
  end
end
luatexbase.registernumber = registernumber
local attributes=setmetatable(
{},
{
__index = function(t,key)
return registernumber(key) or nil
end}
)
luatexbase.attributes = attributes
local attribute_count_name =
                      attribute_count_name or ↲
                            ↳"e@alloc@attribute@count"
local function new_attribute(name)
  tex_setcount("global", attribute_count_name,
                           tex_count[attribute_count_name] + 1)
  if tex_count[attribute_count_name] > 65534 then
    luatexbase_error("No room for a new \\attribute")
  end
  attributes[name]= tex_count[attribute_count_name]
  luatexbase_log("Lua-only attribute " .. name .. " = " ..
                 tex_count[attribute_count_name])
  return tex_count[attribute_count_name]
end
luatexbase.new_attribute = new_attribute
local whatsit_count_name = whatsit_count_name or ↲
      ↳"e@alloc@whatsit@count"
local function new_whatsit(name)
  tex_setcount("global", whatsit_count_name,
                          tex_count[whatsit_count_name] + 1)
  if tex_count[whatsit_count_name] > 65534 then
    luatexbase_error("No room for a new custom whatsit")
  end
  luatexbase_log("Custom whatsit " .. (name or "") .. " = " ..
                 tex_count[whatsit_count_name])
  return tex_count[whatsit_count_name]
end
luatexbase.new_whatsit = new_whatsit
```

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

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

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

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

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

```
}
```

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

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

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

```
local callbacktypes = callbacktypes or {
  find_read_file     = exclusive,
  find_write_file    = exclusive,
  find_font_file     = data,
  find_output_file   = data,
  find_format_file   = data,
  find_vf_file       = data,
  find_map_file      = data,
  find_enc_file      = data,
  find_pk_file       = data,
  find_data_file     = data,
  find_opentype_file = data,
  find_truetype_file = data,
  find_type1_file    = data,
  find_image_file    = data,
  open_read_file     = exclusive,
  read_font_file     = exclusive,
  read_vf_file       = exclusive,
  read_map_file      = exclusive,
  read_enc_file      = exclusive,
  read_pk_file       = exclusive,
  read_data_file     = exclusive,
  read_truetype_file = exclusive,
  read_type1_file    = exclusive,
  read_opentype_file = exclusive,
```

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

```
  find_cidmap_file   = data,
  read_cidmap_file   = exclusive,
  process_input_buffer  = data,
  process_output_buffer = data,
  process_jobname       = data,
  contribute_filter      = simple,
  buildpage_filter       = simple,
  build_page_insert      = exclusive,
  pre_linebreak_filter   = list,
```

```
  linebreak_filter       = list,
  append_to_vlist_filter = exclusive,
  post_linebreak_filter  = list,
  hpack_filter           = list,
  vpack_filter           = list,
  hpack_quality          = list,
  vpack_quality          = list,
  pre_output_filter      = list,
  process_rule           = list,
  hyphenate              = simple,
  ligaturing             = simple,
  kerning                = simple,
  insert_local_par       = simple,
  mlist_to_hlist         = list,

  pre_dump                = simple,
  start_run               = simple,
  stop_run                = simple,
  start_page_number       = simple,
  stop_page_number        = simple,
  show_error_hook         = simple,
  show_warning_message    = simple,
  show_error_message      = simple,
  show_lua_error_hook     = simple,
  start_file              = simple,
  stop_file               = simple,
  call_edit               = simple,
  finish_synctex_callback = simple,

  finish_pdffile = data,
  finish_pdfpage = data,

  define_font           = exclusive,
  glyph_not_found       = exclusive,
  glyph_stream_provider = exclusive,

}
luatexbase.callbacktypes=callbacktypes

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

45.4.5 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.

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

Handler for `data` callbacks.

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

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

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

```
      if ret ~= true then
        alltrue = false
        head = ret
      end
    end
    return alltrue and true or head
  end
end
```

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

Keep a handlers table for indexed access.

```
local handlers = {
  [data]      = data_handler,
  [exclusive] = exclusive_handler,
  [list]      = list_handler,
  [simple]    = simple_handler,
}
```

45.4.6 Public functions for callback management

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

```
local user_callbacks_defaults = { }
```

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

`call_callback`

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

(*End definition for* `call_callback`.)

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

```
local function add_to_callback(name, func, description)
  if not name or name == "" then
    luatexbase_error("Unable to register callback:\n" ..
                     "valid callback name required")
  end
  if not callbacktypes[name] or
    type(func) ~= "function" or
    not description or
    description == "" then
    luatexbase_error(
      "Unable to register callback.\n\n"
        .. "Correct usage:\n"
        .. "add_to_callback(<callback>, <function>, ↲
             ↳<description>)"
    )
  end
  local l = callbacklist[name]
  if l == nil then
    l = { }
    callbacklist[name] = l
    if user_callbacks_defaults[name] == nil then
      callback_register(name, ↲
            ↳handlers[callbacktypes[name]](name))
    end
  end
  local f = {
    func        = func,
```

```
    description = description,
  }
  local priority = #l + 1
  if callbacktypes[name] == exclusive then
    if #l == 1 then
      luatexbase_error(
        "Cannot add second callback to exclusive function\n`" ..
        name .. "'")
    end
  end
  table.insert(l, priority, f)
  luatexbase_log(
    "Inserting `" .. description .. "' at position "
      .. priority .. " in `" .. name .. "'."
  )
end
luatexbase.add_to_callback = add_to_callback
```

(*End definition for* `add_to_callback`.)

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

```
local function remove_from_callback(name, description)
  if not name or name == "" then
    luatexbase_error("Unable to remove function from callback:\n" ..
                     "valid callback name required")
  end
  if not callbacktypes[name] or
    not description or
    description == "" then
    luatexbase_error(
      "Unable to remove function from callback.\n\n"
        .. "Correct usage:\n"
        .. "remove_from_callback(<callback>, <description>)"
    )
  end
  local l = callbacklist[name]
  if not l then
    luatexbase_error(
      "No callback list for `" .. name .. "'\n")
  end
  local index = false
  for i,j in ipairs(l) do
    if j.description == description then
      index = i
      break
    end
  end
  if not index then
    luatexbase_error(
      "No callback `" .. description .. "' registered for `" ..
      name .. "'\n")
```

```
  end
  local cb = l[index]
  table.remove(l, index)
  luatexbase_log(
    "Removing `" .. description .. "' from `" .. name .. "'."
  )
  if #l == 0 then
    callbacklist[name] = nil
    callback_register(name, nil)
  end
  return cb.func,cb.description
end
luatexbase.remove_from_callback = remove_from_callback
```

(*End definition for* `remove_from_callback`.)

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

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

(*End definition for* `in_callback`.)

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

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

List the descriptions of functions registered for the given callback.

`callback_descriptions`

```
local function callback_descriptions (name)
  local d = {}
  if not name
    or name == ""
```

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

uninstall

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

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

```
%</lua>
```

```
%<tex>\catcode`\@=\etatcatcode\relax
```

Latex Classes

phd package and give a background as to how fonts are used in LaTeX.

phd package and give a background as to how fonts are used in LaTeX.

DISSECTING THE BOOK CLASS

This appendix describes the listing of the book class as defined by LaTeX. It is described here with extra commentary in order to enable you to understand, how it all works.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, vo-

lutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

46.1 General

The book class starts with declaring the version of LaTeX, required and naming the class it provides. The class choices are always checked for backward compatibility with the earlier version of LaTeX. All commands that need to be modified between two column and one column layouts, check the setting and branch accordingly. Another primary choice is if the book is to be printed on both sides or only on one side.

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesClass{book}
              [2007/10/19 v1.4h
 Standard LaTeX document class]
```

`\@ptsize` ⟨*dim*⟩

This is set to an empty value at start up. The original idea here was by modifying the value one could scale the text later on. I am not aware of any such usage.

```
\newcommand\@ptsize{}
\newif\if@restonecol
\newif\if@titlepage \@titlepagetrue
\newif\if@openright
\newif\if@mainmatter \@mainmattertrue
```

Paper size. After checking for compatibilty with older versions the code branches to define the different standard paper sizes! The options that are declared are, `a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper` and `executivepaper`. The class will then later on process the options and set the default to `letterpaper`. [1]

```
\if@compatibility\else
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}}
\DeclareOption{a5paper}
   {\setlength\paperheight {210mm}%
    \setlength\paperwidth  {148mm}}
```

[1] The package `geometry`, some classes such as the `Octavo` and `KOMA` classes add additional sizes to cater for other standards.

```
\DeclareOption{b5paper}
   {\setlength\paperheight {250mm}%
    \setlength\paperwidth  {176mm}}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{legalpaper}
   {\setlength\paperheight {14in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{executivepaper}
   {\setlength\paperheight {10.5in}%
    \setlength\paperwidth  {7.25in}}
```

Paper orientation. the paper orientation is set based on the `landscape` option. If it is declared it stores the `\paperheight` into one of the LaTeX kernel scratch registers, `\@tempdima` and then reverses the length with the `\paperwidth`.

```
\DeclareOption{landscape}
   {\setlength\@tempdima   {\paperheight}%
    \setlength\paperheight {\paperwidth}%
    \setlength\paperwidth  {\@tempdima}}
\fi
```

Font sizing The class provides for three font sizes `10pt`, `11pt` and `12pt`. It defaults to ten point text. It then sets the **??**[→ P. ??] accordingly. This is used later on to determine what size of font we are using.

```
\if@compatibility
  \renewcommand\@ptsize{0}
\else
\DeclareOption{10pt}{\renewcommand\@ptsize{0}}
\fi
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
\DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

Recto and verso pages. The class provides the `oneside` and `twoside` options for switching between one side printing or two side printing. It sets the booleans `\if@twoside` and `\if@mparswitch` accordingly. These conditionals are used later to setting other variables.

```
\if@compatibility\else
  \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\fi
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
```

Draft and final options. The options draft and final, just set the `\overfullrule` to either 1pt or 0pt. The `\overfullrule` is a TeX command and simply prints a small vertical line to indicate overfull boxes for the attention of the author.

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\if@compatibility\else
  \DeclareOption{final}{\setlength\overfullrule{0pt}}
\fi
```

Title page option. If the book class, needed such an option is debatable. The `titlepage` option is normally set as true and results in the title being on

its own page. The **notitlepage** will omit the page break and display the title on the same page with that of the opening text. Highly unlikely for any author to use it for a book. It is useful for the article class.

```
\DeclareOption{titlepage}{\@titlepagetrue}
\if@compatibility\else
  \DeclareOption{notitlepage}{\@titlepagefalse}
\fi
```

Display of chapters. Chapters can be set to start only on an even page or any page. The class provides the options **openright** and **openany**.

```
\if@compatibility
  \@openrighttrue
\else
  \DeclareOption{openright}{\@openrighttrue}
  \DeclareOption{openany}{\@openrightfalse}
\fi
```

```
\if@compatibility\else
 \DeclareOption{onecolumn}{\@twocolumnfalse}
\fi
\DeclareOption{twocolumn}{\@twocolumntrue}
\DeclareOption{leqno}{\input{leqno.clo}}
\DeclareOption{fleqn}{\input{fleqn.clo}}
\DeclareOption{openbib}{%
  \AtEndOfPackage{%
   \renewcommand\@openbib@code{%
      \advance\leftmargin\bibindent
      \itemindent -\bibindent
      \listparindent \itemindent
      \parsep \z@
      }%
   \renewcommand\newblock{\par}}%
}
```

At this point all the options have been declared and we can process them, using **??**[→P.??] and **??**[→P.??].

```
\ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
\ProcessOptions
```

The .clo files The book class now inputs the file **.clo** etc that defines the fontsizes for anything specific to the 10pt. These files hold quite a bit of information and size related commands for the standard sizes provided by LaTeX. The **.clo** files also set many other parameters for page sizing, lists, paper sectioning, such as margins, marginpars and the like.

```
\input{bk1\@ptsize.clo}
\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
\renewcommand\baselinestretch{}
\setlength\parskip{0\p@ \@plus \p@}
```

Penalties. Next we set some penalties. The definition of **??**[→P.??], **??**[→P.??] and **??**[→P.??] are in the kernel in the **ltfinal .dtx** file.

```
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
```

Float control parameters. The allowable number of floats on a page are controlled by a number of parameters. These are set here. Many users overwrite these parameters in order to have more control on the placement of floats.

```
\setcounter{topnumber}{2}
\renewcommand\topfraction{.7}
\setcounter{bottomnumber}{1}
\renewcommand\bottomfraction{.3}
\setcounter{totalnumber}{3}
\renewcommand\textfraction{.2}
\renewcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\renewcommand\dbltopfraction{.7}
\renewcommand\dblfloatpagefraction{.5}
```

Running head and foot. A page header or simply header in typography is text which is separated from the main body of text and appears at the top of a printed page. Word processing programs usually provide for the creation and maintenance of page headers, which are often the same from page to page, with merely small differences in information, such as page number.

In publishing, the page header (or "pagehead") is often referred to as the running head. Typical running heads in a book might consist of the book title on the left-hand (verso) page, and the chapter title on the right-hand (recto) page, or chapter title on the verso and subsection title on the recto.

46.2 Running Heads

```
\if@twoside
  \def\ps@headings{%
      \let\@oddfoot\@empty\let\@evenfoot\@empty
      \def\@evenhead{\thepage\hfil\slshape\leftmark}%
      \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
      \let\@mkboth\markboth
 % chapter
  \def\chaptermark##1{%
      \markboth {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \if@mainmatter
            \@chapapp\ \thechapter. \ %
          \fi
        \fi
        ##1}}{}}%
% section
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\z@
          \thesection. \ %
        \fi
        ##1}}}}
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
```

```
    \let\@mkboth\markboth
    \def\chaptermark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \if@mainmatter
            \@chapapp\ \thechapter. \ %
          \fi
        \fi
        ##1}}}}
\fi
\def\ps@myheadings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\@gobbletwo
    \let\chaptermark\@gobble
    \let\sectionmark\@gobble
    }
```

Please note the **??**[→ P. **??**] headings are not defined in the class. These are defined in the LaTeX kernel[2]

```
%\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}
```

Title pages. Title pages are defined between a conditional, that handle the option `titlepage` and. The commands just take mostly of the typography. If you use the option `notitlepage` in the book class, the title will be similar for all practical purposes to that of an `article` and it will appear on the top of the first page.

The `\maketitle` sets the `\footnotesise`, the `\footnoterule` and the `\footnote`.

Titlepage

```
 \if@titlepage
  \newcommand\maketitle{\begin{titlepage}%
  \let\footnotesize\small
  \let\footnoterule\relax
  \let \footnote \thanks
  \null\vfil
  \vskip 60\p@
  \begin{center}%
    {\LARGE \@title \par}%
    \vskip 3em%
    {\large
     \lineskip .75em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
      \vskip 1.5em%
    {\large \@date \par}%       % Set date in \large size.
  \end{center}\par
```

[2] See File J: ltpage.dtx **??**.

```
  \@thanks
  \vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\else
\newcommand\maketitle{\par
  \begingroup
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
    \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
    \long\def\@makefntext##1{\parindent 1em\noindent
            \hb@xt@1.8em{%
                \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
    \if@twocolumn
      \ifnum \col@number=\@ne
        \@maketitle
      \else
        \twocolumn[\@maketitle]%
      \fi
    \else
      \newpage
      \global\@topnum\z@   % Prevents figures from going at top of page.
      \@maketitle
    \fi
    \thispagestyle{plain}\@thanks
  \endgroup
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\def\@maketitle{%
  \newpage
  \null
  \vskip 2em%
```

```
  \begin{center}%
  \let \footnote \thanks
    {\LARGE \@title \par}%
    \vskip 1.5em%
    {\large
      \lineskip .5em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
    \vskip 1em%
    {\large \@date}%
  \end{center}%
  \par
  \vskip 1.5em}
\fi
```

Section counters. In LaTeX all defaults all document section are numbered by default. These numbers are kept in counters, named after the section name. A series of commands are provided to access these numbers. All the counters are in arabic numerals, with the exception of ”part”, which is in Roman.

```
\newcommand*\chaptermark[1]{}
\setcounter{secnumdepth}{2}
\newcounter {part}
\newcounter {chapter}
\newcounter {section}[chapter]
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
\renewcommand \thepart {\@Roman\c@part}
\renewcommand \thechapter {\@arabic\c@chapter}
\renewcommand \thesection {\thechapter.\@arabic\c@section}
\renewcommand\thesubsection    {\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\renewcommand\theparagraph    {\thesubsubsection.\@arabic\c@paragraph}
\renewcommand\thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
\newcommand\@chapapp{\chaptername}
```

Frontmatter, mainmatter and backmatter. These are author command to set mostly, the page numbering and the clearing of pages for two page layouts. Front matter has lower roman pages numbering and the main matter has arabic numerals.

`\frontmatter`

Front matter sets the page opening and teh numbering in Roman lowercase.

```
\newcommand\frontmatter{%
    \cleardoublepage
  \@mainmatterfalse
  \pagenumbering{roman}}
```

\mainmatter

The author command to denote the start of the main body of the publication.

```
\newcommand\mainmatter{%
    \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}}
```

\backmatter

The author command to denote the start of the back matter of the publication. These are still numbered in arabic numerals.

```
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
```

Part. The is the definition of part. The Part is displayed with a plain header and the it goes into the secdef. If the section depth is greater or equal -2, the start counter is increased and the part is added to the toc, using `\addcontentsline`. The partname i.e., default 'Part' gets printer either way except for the star version of the command.

\part

Starts a new part of a book. The star version of the command does not increment the part number.

```
\newcommand\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{plain}%
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
  \null\vfil
  \secdef\@part\@spart}
```

The important command to remember here is **??**$^{\rightarrow \text{P.}\,??}$. This is defined in the kernel and not in the classes `ltsect.dtx`. Essentially in the code `\@part` calls the unstar command and the @spart calls the starred command. We copy the definition from the kernel for convenience.

```
is \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\hstarcmdi[#1]#2{ . . . } % Command to define \chapter[. . . ]{. . . }
3. \def\unstarcmd#1{ . . . } % Command to define \chapter*{. . }
125 \def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

The `@part` starts now,

```
\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >-2\relax
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
    \fi
    \markboth{}{}%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \ifnum \c@secnumdepth >-2\relax
       \huge\bfseries \partname\nobreakspace\thepart
       \par
       \vskip 20\p@
     \fi
     \Huge \bfseries #2\par}%
    \@endpart}
```

The starred version of the command is provided next. The difference the name 'Part" is not displayed. However the parameter provided by the user is displayed. A normal font is provided. Final settings depending on @openright and header styles are set and the code macro is completed.

```
\def\@spart#1{%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \Huge \bfseries #1\par}%
    \@endpart}

\def\@endpart{\vfil\newpage
              \if@twoside
               \if@openright
                \null
                \thispagestyle{empty}%
                \newpage
               \fi
              \fi
              \if@tempswa
                \twocolumn
              \fi}
```

Chapter . The chapter definition follows, the same pattern as that of the part definitions. It calls `\secdef` and defines commands for the starred and unstarred versions.

```
\newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                    \thispagestyle{plain}%
                    \global\@topnum\z@
                    \@afterindentfalse
                    \secdef\@chapter\@schapter}
```

Unstarred version

```
\def\@chapter[#1]#2{
    \ifnum \c@secnumdepth >\m@ne
       \if@mainmatter
             \refstepcounter{chapter}%
             \typeout{\@chapapp\space\thechapter.}%
             \addcontentsline{toc}{chapter}%
             {\protect\numberline{\thechapter}#1}%
       \else
             \addcontentsline{toc}{chapter}{#1}%
      \fi
   \else
         \addcontentsline{toc}{chapter}{#1}%
   \fi
   \chaptermark{#1}%
   \addtocontents{lof}{\protect\addvspace{10\p@}}%
   \addtocontents{lot}{\protect\addvspace{10\p@}}%
   \if@twocolumn
        \@topnewpage[\@makechapterhead{#2}]%
   \else
        \@makechapterhead{#2}%
        \@afterheading
   \fi}
```

Defining the looks of the Chapter heading. Good practice dictates, that when you change the chapterhead layout for the numbered version, you also change it for the star version of the command. You can do that by using two different macros, although at first glance it might be difficult to see where the difference is.

```
\def\@makechapterhead
\def\@makeschapterhead
```

```
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
      \fi
    \fi
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
```

```
    \vskip 40\p@
  }}
```

Finally the starred version of the command is called. This now checks for twocolumn or one column via an if sttaement and executes, the makeschapterhead. Another mysterious and wonderful command appears again from the LaTeX source2e.`\@afterheading`. This command is just a hook for custom headings? (Needs to be reviewed again).

```
\def\@schapter#1{\if@twocolumn
                    \@topnewpage[\@makeschapterhead{#1}]%
                  \else
                    \@makeschapterhead{#1}%
                    \@afterheading
                  \fi}
```

And finally the `\@makeschapterhead` (remember **s** for **s**tar).

```
\def\@makeschapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright
    \normalfont
    \interlinepenalty\@M
    \Huge \bfseries  #1\par\nobreak
    \vskip 40\p@
  }}
```

All sorts of variations of the above two commands can be found in different classes, such as `KOMA`, `memoir` and others. The example which follows, typesets the headings as shown in Figure **??**. The `@makechapterhead` command is modified to produce a centered heading which is displayed between two heavy rules. This style can be found in quite a number of books.

46.3 Full working example

```
\documentclass[oneside]{book}
\usepackage[english]{babel}
\usepackage{lipsum}
\makeatletter
\def\thickhrule{\leavevmode \leaders \hrule height 1ex ⤸
      ⤹\hfill \kern \z@}

%% Note the difference between the commands the one is
%% make and the other one is makes
\renewcommand{\@makechapterhead}[1]{%
  \vspace*{10\p@}%
  {\parindent \z@ \centering \reset@font
        {\Huge \scshape  \thechapter }
        \par\nobreak
        \vspace*{10\p@}%
        \interlinepenalty\@M
        \thickhrule
        \par\nobreak
        \vspace*{2\p@}%
        {\Huge \bfseries #1\par\nobreak}
        \par\nobreak
```

Figure 46.1: Modifying the way the chapterhead looks can be achieved by redefining the `\@makechapterhead` and `\@makeschapterhead` commands.

```
        \vspace*{2\p@}%
        \thickhrule
    \vskip 40\p@
    \vskip 100\p@
  }}

%% This is makes
\def\@makeschapterhead#1{%
  \vspace*{10\p@}%
  {\parindent \z@ \centering \reset@font
        {\Huge \scshape \vphantom{\thechapter}}
        \par\nobreak
        \vspace*{10\p@}%
        \interlinepenalty\@M
        \thickhrule
        \par\nobreak
        \vspace*{2\p@}%
        {\Huge \bfseries #1\par\nobreak}
        \par\nobreak
        \vspace*{2\p@}%
        \thickhrule
    \vskip 100\p@
  }}
\begin{document}
\chapter{The Real Numbers}
\lipsum[1-2]
\chapter*{The Imaginary Numbers}
```

```
\lipsum[1-2]
\end{document}
```

The sections. In this section, all the document elements besides the Chapter and the Part are Defined. They use the mother of all commands from the kernel ltsection.dtx, named **??** → P. ??. This is just a call to the kernel command. No other settings are done here. In order to remember what it does we refer to its definition in the kernel. Of interest is the sixth argument which sets the font style.

The parameter takes eight parameters, some of them optional. We discuss this command in more detail in the kernel chapter.

```
\newcommand\section{\@startsection {section}{1}{\z@}%
                                   {-3.5ex \@plus -1ex ↩
                                        ↩\@minus -.2ex}%
                                   {2.3ex \@plus.2ex}%
                                   {\normalfont\Large\bfseries}}
\newcommand\subsection{\@startsection{subsection}{2}{\z@}%
                                     {-3.25ex\@plus -1ex ↩
                                          ↩\@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\large\bfseries}}
\newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                     {-3.25ex\@plus -1ex ↩
                                          ↩\@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\normalsize\bfseries}}
\newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
                                    {3.25ex \@plus1ex ↩
                                         ↩\@minus.2ex}%
                                    {-1em}%
                                    {\normalfont\normalsize\bfseries}}
\newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                       {3.25ex \@plus1ex ↩
                                            ↩\@minus .2ex}%
                                       {-1em}%
                                      {\normalfont\normalsize\bfseries}}
```

46.4 Lists

```
\if@twocolumn
  \setlength\leftmargini  {2em}
\else
  \setlength\leftmargini  {2.5em}
\fi
\leftmargin  \leftmargini
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv  {1.7em}
\if@twocolumn
  \setlength\leftmarginv  {.5em}
  \setlength\leftmarginvi {.5em}
\else
  \setlength\leftmarginv  {1em}
  \setlength\leftmarginvi {1em}
```

```
\fi
\setlength  \labelsep  {.5em}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}
```

46.5 The description Environment

`\descriptionlabel`{⟨*text*⟩}

```
\newenvironment{description}
               {\list{}{\labelwidth\z@ \itemindent-\leftmargin
                        \let\makelabel\descriptionlabel}}
               {\endlist}

\newcommand*\descriptionlabel[1]{\hspace\labelsep
                                 \normalfont\bfseries #1}
```

46.6 Verse Environment

The verse environment LaTeX's `verse` environment, can only serve for the incidental use of a few stanzas. It leaves most of the formatting to the author. It redefines the line break `\\` to a `\centercr`.

```
\newenvironment{verse}
    {\let\\\@centercr[3]
     \list{}{\itemsep  \z@
             \itemindent   -1.5em%
             \listparindent\itemindent
             \rightmargin  \leftmargin
             \advance\leftmargin 1.5em}%
       \item\relax}
    {\endlist}
```

[3] This is defined in ltmiscen.dtx

```
\begin{Verse}
   My mobile test\\
   this is other\\
   this is last\\
\end{Verse}
```

The environment doesn't really do much, the way I see it but just move the poem a couple of ems inwards to much the definition of lists. Most people will want more from a poem environment.

> My mobile test
> this is other
> this is last

The simplest thing we can add to this environment if we want to modify it, is a hook. This we can do using the **blckcntrl** package. [4].

```
\renewenvironment{verse}
50 {\let\\\@centercr
51 \relax\list{}{\setlength{\itemsep}{\z@}%
52 \setlength{\itemindent}{-1.5em}%
53 \setlength{\listparindent}{\itemindent}%
54 \setlength{\rightmargin}{\leftmargin}%
55 \addtolength{\leftmargin}{1.5em}}%
56 \item\relax\PreVerse\relax}
57 {\endlist}
```

Using the command **\PreVerse**, we can add a block at the beginning of the block. For example some code to make a poem title and insert it later on. The setting of the rightmargin to the leftmargin here is curious. It might for example give us problems with **tufte-latex** classes.

46.7 Quotation and quote environments

The quote and quotation environments. The environments **quote** and **quotation** are defined next. Again they are defined using the general **??** → P. ?? environment. Again the general **\list**, is used in the definition. The dimension **??** → P. ?? is set to 1.5 em. (See Kernel file **ltlists.dtx ??**).

```
\newenvironment{quotation}
                {\list{}{\listparindent 1.5em%
                         \itemindent\listparindent
                         \rightmargin\leftmargin
                         \parsep\z@ \@plus\p@}%
                 \item\relax}
                {\endlist}
```

```
\newenvironment{quote}
                {\list{}{\rightmargin\leftmargin}%
                 \item\relax}
                {\endlist}
```

[4] From the http://www.ifi.uio.no/it/latex-links/blkcntrl.pdf

```

\section{The \protect\texttt{titlepage} environment}
\if@compatibility
\newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\z@
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
    }
\else
\newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\@ne
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
     \if@twoside\else
        \setcounter{page}\@ne
     \fi
    }
\fi
```

Appendix B

Experimental Errors

B.1 First section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

B.1.1 First subsection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5

The Appendix. Similarly to the chapter sectioning commands, the Appendix is not defined as a section. It simply sets the chapter and section counters to zero and sets the name of the section. All the relevant counters and uses letters for the numbering of the following chapters etc. If you closely follow the code, it is all based on the chapter command, except that it defaults to Alphanumeric counting.

```
\appendix
```

```
\newcommand\appendix{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}[5]
  \gdef\thechapter{\@Alph\c@chapter}}
```

An Appendix page has the same looks and feel to that of a Chapter. For all practical purposes, it is a chapter, with different labels and uppecase letters numbering.

General Settings. Here, some general settings are set. These include settings for framed boxes, tabbing separators and array column separators.

```
\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}
\setlength\tabbingsep{\labelsep}
\skip\@mpfootins = \skip\footins
```

The `\fboxsep` and `\fboxrule` are then set to `3pt` and `.4pt` respectively. These are used with frame boxes.

```
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
```

Equation numbering The equation counter is reset according to the chapter counter, using the LaTeX kernel command `\@addtoreset`.

```
\@addtoreset {equation}{chapter}
\renewcommand\theequation
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

46.8 FIGURE AND TABLE ENVIRONMENTS

Figure Environment The figure environment is defined using commands that have been provided by the kernel. The command `\thefigure` is first redefined to display the combination of the chapter dot figure counter, all in arabic numerals. The extension for the list of figures and finally the floats for single column and double column.

```
\newcounter{figure}[chapter]
\renewcommand \thefigure
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
```

[5]The actual literal used for \appendixname is defined later on, so that you can customize the language

Table Environment Table floats are defined the same way like the figures with their respective counters and names.

```
\newcounter{table}[chapter]
\renewcommand \thetable
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\nobreakspace\thetable}


\newenvironment{table}
               {\@float{table}}
               {\end@float}

\newenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

Captions The captioning macros are rather short but need a bit of explanation. First some lengths are defined. The lengths are for `abovecaptionskip` and `belowcaptionskip` are set equal to a default of 10pt as for the font-size, but the length `belowcaptionskip` is set to `opt`.

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{10\p@}
\setlength\belowcaptionskip{0\p@}

\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1: #2}
  \ifdim \wd\@tempboxa >\hsize
    #1: #2\par
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
```

The `@makecaption` macro is also interesting. Firstly note in line the use of a colon (:). So if you do not like to have this you know where you need to go and change it. The contents of the caption are first saved into a box. If the box is greater than `hsize` then they are written like a paragraph otherwise, they are centered. Note that the centering is done using `\hfil\box\@tempoxa\hfil`. The mysterious command `\hb@xt` is defined in the kernel and is equivalent to `\hbox to`.

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

It is simply an abbreviation of `\hbox to`. There are many shortcut commands like this, so the command just again sets the caption in a horizontal box. There is more to the story later on.

46.9 Defining the old style font commands

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

46.10 Table of contents

Firstly we define the width of the box that the page number is set. Use ems so that it does not need to be redefined for every change in font size. Toc entries are treated as rectangular areas where the text and probably a filler will be written. Let's draw such an area (of course, the lines themselves are not printed):

The space between the left page margin and the left edge of the area will be named `<left>`; similarly we have `<right>`. You are allowed to modify the beginning of the first line and the ending of the last line. For example by "taking up" both places with `\hspace*{2pc}` the area becomes:

And by "clearing" space in both places with `\hspace*{-2pc}` the area becomes:

If you have seen tocs, the latter should be familiar to you– the label at the very beginning, the page at the very end:

```
3.2  This is an example showing that toc
     entries fits in that scheme . . . .   4
```

```
\newcommand\@pnumwidth{1.55em}%Width of box in which page number is set.
```

We then define the margin and the dotsep. We also set the toc counter to whatever is require (don't go too deep especially if you have an index).

```
\newcommand\@tocrmarg{2.55em}%Right margin indentation for all but last line of multiple-line entries.
\newcommand\@dotsep{4.5}%Separation between dots, in mu units. Should be \def'd to a number like
2 or 1.7
```

\tableofcontents

Defining the contents table. The author is provided with the author command `\tableofcontents`. All format information is provided at this point.

```
\setcounter{tocdepth}{2}
\newcommand\tableofcontents{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\contentsname
        \@mkboth{%
           \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%
    \@starttoc{toc}%
    \if@restonecol\twocolumn\fi
    }
```

```
\newcommand*\l@part[2]{%
  \ifnum \c@tocdepth >-2\relax
    \addpenalty{-\@highpenalty}%
    \addvspace{2.25em \@plus\p@}%
    \setlength\@tempdima{3em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      {\leavevmode
       \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
       \nobreak
         \global\@nobreaktrue
         \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
```

```

\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip 1.0em \@plus\p@
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
```

The five remaining levels (entry in LaTeX terminology, are defined next). This is done with the general LaTeX kernel command

```
\@dottedtocline{<level>}{<indent>}{<numwidth>}{<title>}{<page>}: ↲
      ↳Macro
to produce a table of contents line with the following ↲
      ↳parameters:
```

The commands for the remaining sections are defined as follows:

```
\newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
```

So where are the last two parameters? These are just zeroed here!

I can assure that the **dotted** type of section bothers a lot of people. Most new books will both compact the table of contents as well as remove the dots. You can use the **titlesec** and **titletoc** to do this rather than redefining the kernel commands or the standard classes styles.

46.10.1 List of figures, tables etc

```
\newcommand\listoffigures{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listfigurename}%
      \@mkboth{\MakeUppercase\listfigurename}%
              {\MakeUppercase\listfigurename}%
    \@starttoc{lof}%
    \if@restonecol\twocolumn\fi
    }
```

The interesting command here is the `@starttoc{lof}`. This simply does all the housekeeping to open a file. as you can see it is not too difficult to have file extension names other than the standard ones.

The `l@` commands for the Table of Contents are defined as per the rest of the sectioning commands.

```
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand\listoftables{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listtablename}%
      \@mkboth{%
          \MakeUppercase\listtablename}%
         {\MakeUppercase\listtablename}%
    \@starttoc{lot}%
    \if@restonecol\twocolumn\fi
    }
\let\l@table\l@figure
```

46.11 Bibliographies

LaTeX provides some basic bibliographic commands. Every entry is defined to be displayed in a block. It starts by defining a new length `\bibindent`. Entries are displayed using the **??** →P. ??. The commands here are mainly to set parameters for macros already provide by the kernel.

`\begin{thebibliography}`{⟨*widest label*⟩}
⟨*environment content*⟩
`\end{thebibliography}`

```
\newdimen\bibindent
\setlength\bibindent{1.5em}
\newenvironment{thebibliography}[1]
     {\chapter*{\bibname}%
      \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
      \list{\@biblabel{\@arabic\c@enumiv}}%
           {\settowidth\labelwidth{\@biblabel{#1}}%
            \leftmargin\labelwidth
            \advance\leftmargin\labelsep
            \@openbib@code
            \usecounter{enumiv}%
            \let\p@enumiv\@empty
            \renewcommand\theenumiv{\@arabic\c@enumiv}}%
      \sloppy
      \clubpenalty4000
      \@clubpenalty \clubpenalty
      \widowpenalty4000%
      \sfcode`\.\@m}
     {\def\@noitemerr
       {\@latex@warning{Empty `thebibliography' environment}}%
      \endlist}
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
\let\@openbib@code\@empty
```

46.12 The Index Environment

This is a short environment definition for styling the Index. It defines in line [**??**] the `@idxidtem`, which is then used to define `\subitem` and `\subsubitem` styling.

```
\newenvironment{theindex}
   {\if@twocolumn
      \@restonecolfalse
      \else
         \@restonecoltrue
      \fi
      \twocolumn[\@makeschapterhead{\indexname}]%
      \@mkboth{\MakeUppercase\indexname}%
              {\MakeUppercase\indexname}%
                \thispagestyle{plain}\parindent\z@
                \parskip\z@ \@plus .3\p@\relax
                \columnseprule \z@
                \columnsep 35\p@
                \let\item\@idxitem}
      {\if@restonecol\onecolumn\else\clearpage\fi}
\newcommand\@idxitem{\par\hangindent 40\p@}
\newcommand\subitem{\@idxitem \hspace*{20\p@}}
\newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
\newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ ↩
      ↪\@minus3\p@\relax}
```

46.13 Footnotes

`\footnoterule`

Footnote rules. Footnote rules are defined by renewing the command `\footnoterule`. Counters for footnotes are reset based on the chapter counters. The footnote command `\@makefntext` provides the formatting. It also gives the user the ability to use these to insert footnotes, in difficult places.

```
\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}

\@addtoreset{footnote}{chapter}
```

`\@makefntext`

`\@makefnmark`

```
\newcommand\@makefntext[1]{%
    \parindent 1em%
    \noindent
    \hb@xt@1.8em{\hss\@makefnmark}#1}
```

46.14 Catering for Other Languages

Structural element names. LaTeX does not provide by itself the means to change the structural element names to a language other than English. Howerer, their names are defined in a series of commands, that make it easier to be overwritten to change them to another language. As they are separate from the macros that use them, it is easy to overwrite them, in order to use another language. This is what the Babel package does. Note the Section, is not defined here.

`\contentsname`

Typesets the contents name.

```
\newcommand\contentsname{Contents}
```

`\listfigurename`

Typesets "List of Figures", if it is not redefined.

```
\newcommand\listfigurename{List of Figures}
\newcommand\listtablename{List of Tables}
\newcommand\bibname{Bibliography}
\newcommand\indexname{Index}
\newcommand\figurename{Figure}
\newcommand\tablename{Table}
\newcommand\partname{Part}
\newcommand\chaptername{Chapter}
\newcommand\appendixname{Appendix}
```

Dates Not much of a use but the month names are also defined here in an `\ifcase` statement. Again they can be overwritten by Babel.

`\today`

```
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

Multicolumn gutter and rule. Here two lengths are set. The distance between two columns of text and the width of the separating rule.

```
\setlength\columnsep{10\p@}
\setlength\columnseprule{0\p@}
```

46.15 Final

```
\pagestyle{headings}
\pagenumbering{arabic}
\if@twoside
\else
```

```
  \raggedbottom
\fi
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
\fi
\endinput
%%
%% End of file `book.cls'.
```

46.16 Ending remarks

It is to the credit of Lamport and his associates that he was the first one to produce a system of mark-up that structured documents, using the TeX typographical engine. The class is widely used and many variants exist. One area that can be improved is to provide more 'hooks' to enable programmers to redefine classes more easily.

Since the class has been published new packages have established themselves as the 'de facto' standards of defining portions of the class. For example the nonew class will attempt to define all the papers as Lamport did, but would rather use the **geometry** package to do so. Top and bottom headings are defined using the **fancyverb**.

> It was when the code was written, but is not now (in my opinion). The current LaTeX2e kernel was release in 1992 and carries forward a lot of material from LaTeX2.09. Even with these optimizations and the old 'autoload' system, there were a lot of systems that LaTeX was too big for on release. So looked at in the early 1990s this was entirely sensible.
>
> I'd say this is no longer needed as in most LaTeX documents today there are a lot of tokens used by things like pgf which make the modest saving in optimisation pretty meaningless. One of the things we're doing in LaTeX3 is trying to move to more logical constructs at the expense of efficiency in tokens, at least at a higher level. (Right at the core of expl3 there is still a need to watch the number of expansions, etc., and this is an area where we may yet need some more optimisation.)

You can think of the LaTeX classes working at three levels.

1. Selecting paper sizes and defining main page elements.
2. They define how the document is section. I have called this sectioning by referring to it as structural commands.
3. It provides the typesetting of these structural elements.

Unfortunately, they are not separated in a way that makes it easy for them to be modified. A plethora of packages assists the author in modifying every

type of sectioning and formatting decisions of Lamport. Most authors will focus on the formatting commands. Some will add a bit of structure, perhaps some special sections for questions and answers. If you have used the titlesec package for modifying the sections, the caption package for modifying the way captions are displayed, the fancyhdr for headers, the titletoc for the way table of contents are displayed, one of the bibliography packages what begs to be question is what remains? Very little. You might as well at this point decide on a new class. It will be more efficient and you will have better control. Separation of structure from presentational decisions is important. Some common structural elements that are missing should be integrated in. The KOMA classes and memoir went totally overboard, in that they try to be everything to everybody. A system that is nearer to defining a structural template and then decorate it with a selction of fonts, colors, spacing and the like would have been more appropriate.

46.17 The .clo files

```
%%
%% This is file `bk10.clo',

\ProvidesFile{bk10.clo}
              [2007/10/19 v1.4h
      Standard LaTeX file (size option)]
```

??[→ P. ??] = ⟨*commands*⟩ Redefined for each font-size option, 10pt, 11pt, 12pt etc.

```
\renewcommand\normalsize{%
   \@setfontsize\normalsize\@xpt\@xiipt
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \belowdisplayskip \abovedisplayskip
   \let\@listi\@listI}
\normalsize
\newcommand\small{%
   \@setfontsize\small\@ixpt{11}%
   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus2\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip
}
\newcommand\footnotesize{%
   \@setfontsize\footnotesize\@viiipt{9.5}%
   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus\p@
   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 3\p@ \@plus\p@ \@minus\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
```

```
                \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip
}
\newcommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\newcommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\newcommand\large{\@setfontsize\large\@xiipt{14}}
\newcommand\Large{\@setfontsize\Large\@xivpt{18}}
\newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
\newcommand\huge{\@setfontsize\huge\@xxpt{25}}
\newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
```

Paragraph indentation. This parameter is controlled by the TEX command `parindent`. It is set narrower in two column text, to avoid problems with hyphenation that can result in overfull boxes.

```
\if@twocolumn
  \setlength\parindent{1em}
\else
  \setlength\parindent{15\p@}
\fi

\setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
\setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
\setlength\headheight{12\p@}
\setlength\headsep   {.25in}
\setlength\topskip   {10\p@}
\setlength\footskip{.35in}
\if@compatibility \setlength\maxdepth{4\p@} \else
\setlength\maxdepth{.5\topskip} \fi
\if@compatibility
  \if@twocolumn
    \setlength\textwidth{410\p@}
  \else
    \setlength\textwidth{4.5in}
  \fi
\else
  \setlength\@tempdima{\paperwidth}
  \addtolength\@tempdima{-2in}
  \setlength\@tempdimb{345\p@}
  \if@twocolumn
    \ifdim\@tempdima>2\@tempdimb\relax
      \setlength\textwidth{2\@tempdimb}
    \else
      \setlength\textwidth{\@tempdima}
    \fi
  \else
    \ifdim\@tempdima>\@tempdimb\relax
      \setlength\textwidth{\@tempdimb}
    \else
      \setlength\textwidth{\@tempdima}
    \fi
  \fi
\fi
\if@compatibility\else
```

```
  \@settopoint\textwidth
\fi
\if@compatibility
  \setlength\textheight{41\baselineskip}
\else
  \setlength\@tempdima{\paperheight}
  \addtolength\@tempdima{-2in}
  \addtolength\@tempdima{-1.5in}
  \divide\@tempdima\baselineskip
  \@tempcnta=\@tempdima
  \setlength\textheight{\@tempcnta\baselineskip}
\fi
\addtolength\textheight{\topskip}
\if@twocolumn
 \setlength\marginparsep {10\p@}
\else
  \setlength\marginparsep{7\p@}
\fi
\setlength\marginparpush{5\p@}
\if@compatibility
   \setlength\oddsidemargin   {.5in}
   \setlength\evensidemargin  {1.5in}
   \setlength\marginparwidth {.75in}
  \if@twocolumn
     \setlength\oddsidemargin  {30\p@}
     \setlength\evensidemargin {30\p@}
     \setlength\marginparwidth {48\p@}
  \fi
\else
  \if@twoside
    \setlength\@tempdima        {\paperwidth}
    \addtolength\@tempdima      {-\textwidth}
    \setlength\oddsidemargin    {.4\@tempdima}
    \addtolength\oddsidemargin  {-1in}
    \setlength\marginparwidth   {.6\@tempdima}
    \addtolength\marginparwidth {-\marginparsep}
    \addtolength\marginparwidth {-0.4in}
  \else
    \setlength\@tempdima        {\paperwidth}
    \addtolength\@tempdima      {-\textwidth}
    \setlength\oddsidemargin    {.5\@tempdima}
    \addtolength\oddsidemargin  {-1in}
    \setlength\marginparwidth   {.5\@tempdima}
    \addtolength\marginparwidth {-\marginparsep}
    \addtolength\marginparwidth {-0.4in}
    \addtolength\marginparwidth {-.4in}
  \fi
  \ifdim \marginparwidth >2in
     \setlength\marginparwidth{2in}
  \fi
  \@settopoint\oddsidemargin
  \@settopoint\marginparwidth
  \setlength\evensidemargin  {\paperwidth}
  \addtolength\evensidemargin{-2in}
```

```
  \addtolength\evensidemargin{-\textwidth}
  \addtolength\evensidemargin{-\oddsidemargin}
  \@settopoint\evensidemargin
\fi
```

Top margin Next the top margin is calculated. In earlier versions the **??** → P. ?? was a fixed number. In this class, it is automatically calculated form the `\paperheight` (as the user only inputs the papersize through one of the paper selection options).

```
\if@compatibility
  \setlength\topmargin{.75in}
\else
  \setlength\topmargin{\paperheight}
  \addtolength\topmargin{-2in}
  \addtolength\topmargin{-\headheight}
  \addtolength\topmargin{-\headsep}
  \addtolength\topmargin{-\textheight}
  \addtolength\topmargin{-\footskip}     % this might be ↩
       ↪wrong! (previously set at 0.35in)
  \addtolength\topmargin{-.5\topmargin}
  \@settopoint\topmargin
\fi
```

The lists settings follow. Similarly all values are hard-coded based on the font size.

```
\setlength\footnotesep{6.65\p@}
\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
```

`\listi`

```
\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
\def\@listi{\leftmargin\leftmargini
            \parsep 4\p@ \@plus2\p@ \@minus\p@
            \topsep 8\p@ \@plus2\p@ \@minus4\p@
            \itemsep4\p@ \@plus2\p@ \@minus\p@}
\let\@listI\@listi
\@listi
\def\@listii {\leftmargin\leftmarginii
              \labelwidth\leftmarginii
              \advance\labelwidth-\labelsep
              \topsep    4\p@ \@plus2\p@ \@minus\p@
              \parsep    2\p@ \@plus\p@  \@minus\p@
              \itemsep   \parsep}
\def\@listiii{\leftmargin\leftmarginiii
```

```
               \labelwidth\leftmarginiii
               \advance\labelwidth-\labelsep
               \topsep    2\p@ \@plus\p@\@minus\p@
               \parsep    \z@
               \partopsep \p@ \@plus\z@ \@minus\p@
               \itemsep   \topsep}
\def\@listiv {\leftmargin\leftmarginiv
               \labelwidth\leftmarginiv
               \advance\labelwidth-\labelsep}
\def\@listv  {\leftmargin\leftmarginv
               \labelwidth\leftmarginv
               \advance\labelwidth-\labelsep}
\def\@listvi {\leftmargin\leftmarginvi
               \labelwidth\leftmarginvi
               \advance\labelwidth-\labelsep}

%%
%% End of file `bk10.clo'.
```

phd package and give a background as to how fonts are used in LaTeX.

```
%<article|report|book>\NeedsTeXFormat{LaTeX2e}[1995/12/01]
%<article>\ProvidesClass{article}
%<report>\ProvidesClass{report}
%<book>\ProvidesClass{book}
%<10pt&!bk>\ProvidesFile{size10.clo}
%<11pt&!bk>\ProvidesFile{size11.clo}
%<12pt&!bk>\ProvidesFile{size12.clo}
%<10pt&bk>\ProvidesFile{bk10.clo}
%<11pt&bk>\ProvidesFile{bk11.clo}
%<12pt&bk>\ProvidesFile{bk12.clo}
%<*driver>
\ProvidesFile{classes.drv}
%</driver>
              [2014/09/29 v1.4h
%<article|report|book> Standard LaTeX document class]
%<10pt|11pt|12pt>      Standard LaTeX file (size option)]
```

47.1 A driver for this document

The next bit of code contains the documentation driver file for TEX, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

47.2 The DOCSTRIP modules

The following modules are used in the implementation to direct DOCSTRIP in generating the external files:

article	produce the documentclass article
report	produce the documentclass report
size10	produce the class option for 10pt
size11	produce the class option for 11pt
size12	produce the class option for 12pt
book	produce the documentclass book
bk10	produce the book class option for 10pt
bk11	produce the book class option for 11pt
bk12	produce the book class option for 12pt
driver	produce a documentation driver file

47.3 Initial Code

In this part we define a few commands that are used later on.

`\@ptsize` This control sequence is used to store the second digit of the pointsize we are typesetting in. So, normally, it's value is one of 0, 1 or 2.

```
%<*article|report|book>
\newcommand\@ptsize{}
```

(*End definition for* `disable_callback` *and others.*)

```
\newif\if@restonecol
\newif\if@titlepage
%<article>\@titlepagefalse
%<!article>\@titlepagetrue
%<!article>\newif\if@openright
```

\if@mainmatter The switch **\if@mainmatter**, only available in the document class book, indicates whether we are processing the main material in the book.

```
%<book>\newif\if@mainmatter \@mainmattertrue
```

(*End definition for* `\if@mainmatter`.)

47.4 Declaration of Options

47.4.1 Setting Paper Sizes

The variables **\paperwidth** and **\paperheight** should reflect the physical paper size after trimming. For desk printer output this is usually the real paper size since there is no post-processing. Classes for real book production will probably add other paper sizes and additionally the production of crop marks for trimming. In compatibility mode, these (and some of the subsequent) options are disabled, as they were not present in LaTeX2.09.

```
\if@compatibility\else
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}}
\DeclareOption{a5paper}
   {\setlength\paperheight {210mm}%
    \setlength\paperwidth  {148mm}}
\DeclareOption{b5paper}
   {\setlength\paperheight {250mm}%
    \setlength\paperwidth  {176mm}}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{legalpaper}
   {\setlength\paperheight {14in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{executivepaper}
   {\setlength\paperheight {10.5in}%
    \setlength\paperwidth  {7.25in}}
\DeclareOption{landscape}
   {\setlength\@tempdima   {\paperheight}%
    \setlength\paperheight {\paperwidth}%
    \setlength\paperwidth  {\@tempdima}}
\fi
\if@compatibility
  \renewcommand\@ptsize{0}
\else
\DeclareOption{10pt}{\renewcommand\@ptsize{0}}
```

```
\fi
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
\DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

```
\if@compatibility\else
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\fi
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
```

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\if@compatibility\else
\DeclareOption{final}{\setlength\overfullrule{0pt}}
\fi
```

```
\DeclareOption{titlepage}{\@titlepagetrue}
\if@compatibility
  \else
  \DeclareOption{notitlepage}{\@titlepagefalse}
\fi
```

```
%<!article>\if@compatibility
%<book>\@openrighttrue
%<!article>\else
%<!article>\DeclareOption{openright}{\@openrighttrue}
%<!article>\DeclareOption{openany}{\@openrightfalse}
%<!article>\fi
```

```
\if@compatibility\else
\DeclareOption{onecolumn}{\@twocolumnfalse}
\fi
\DeclareOption{twocolumn}{\@twocolumntrue}
```

```
\DeclareOption{leqno}{\input{leqno.clo}}
```

```
\DeclareOption{fleqn}{\input{fleqn.clo}}
```

47.4.2 Open bibliography

The option `openbib` produces the "open" bibliography style, in which each block starts on a new line, and succeeding lines in a block are indented by `\bibindent`.

```
\DeclareOption{openbib}{%
```

First some hook into the bibliography environment is filled.

```
  \AtEndOfPackage{%
   \renewcommand\@openbib@code{%
      \advance\leftmargin\bibindent
      \itemindent -\bibindent
      \listparindent \itemindent
      \parsep \z@
      }%
```

```
   \renewcommand\newblock{\par}}%
}
```

```
%<*article>
\ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
```

```
%</article>
%<*report>
\ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openany}
%</report>
%<*book>
\ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
%</book>
```

The **\ProcessOptions** command causes the execution of the code for every option FOO which is declared and for which the user typed the FOO option in his **\documentclass** command. For every option BAR he typed, which is not declared, the option is assumed to be a global option. All options will be passed as document options to any **\usepackage** command in the document preamble.

```
\ProcessOptions
```

```
%<!book>\input{size1\@ptsize.clo}
%<book>\input{bk1\@ptsize.clo}
%</article|report|book>
```

```
%<*10pt|11pt|12pt>
\renewcommand\normalsize{%
%<*10pt>
   \@setfontsize\normalsize\@xpt\@xiipt
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
%</10pt>
%<*11pt>
   \@setfontsize\normalsize\@xipt{13.6}%
   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
%</11pt>
%<*12pt>
   \@setfontsize\normalsize\@xiipt{14.5}%
   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
%</12pt>
```

```
   \belowdisplayskip \abovedisplayskip
   \let\@listi\@listI}
```

```
\normalsize
```

```
\newcommand\small{%
%<*10pt>
   \@setfontsize\small\@ixpt{11}%
   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus2\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
%</10pt>
```

```
%<*11pt>
   \@setfontsize\small\@xpt\@xiipt
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 6\p@ \@plus2\p@ \@minus2\p@
               \parsep 3\p@ \@plus2\p@ \@minus\p@
               \itemsep \parsep}%
%</11pt>
%<*12pt>
   \@setfontsize\small\@xipt{13.6}%
   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 9\p@ \@plus3\p@ \@minus5\p@
               \parsep 4.5\p@ \@plus2\p@ \@minus\p@
               \itemsep \parsep}%
%</12pt>
   \belowdisplayskip \abovedisplayskip
}
\newcommand\footnotesize{%
%<*10pt>
   \@setfontsize\footnotesize\@viiipt{9.5}%
   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus\p@
   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 3\p@ \@plus\p@ \@minus\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
%</10pt>
%<*11pt>
   \@setfontsize\footnotesize\@ixpt{11}%
   \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
%</11pt>
%<*12pt>
   \@setfontsize\footnotesize\@xpt\@xiipt
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 6\p@ \@plus2\p@ \@minus2\p@
               \parsep 3\p@ \@plus2\p@ \@minus\p@
               \itemsep \parsep}%
%</12pt>
   \belowdisplayskip \abovedisplayskip
```

```
}
%</10pt|11pt|12pt>
```

\scriptsize
\tiny
\large
\Large
\LARGE
\huge
\Huge

These are all much simpler than the previous macros, they just select a new fontsize, but leave the parameters for displays and lists alone.

```
%<*10pt>
\newcommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\newcommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\newcommand\large{\@setfontsize\large\@xiipt{14}}
\newcommand\Large{\@setfontsize\Large\@xivpt{18}}
\newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
\newcommand\huge{\@setfontsize\huge\@xxpt{25}}
\newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
%</10pt>
%<*11pt>
\newcommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
\newcommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
\newcommand\large{\@setfontsize\large\@xiipt{14}}
\newcommand\Large{\@setfontsize\Large\@xivpt{18}}
\newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
\newcommand\huge{\@setfontsize\huge\@xxpt{25}}
\newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
%</11pt>
%<*12pt>
\newcommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
\newcommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
\newcommand\large{\@setfontsize\large\@xivpt{18}}
\newcommand\Large{\@setfontsize\Large\@xviipt{22}}
\newcommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
\newcommand\huge{\@setfontsize\huge\@xxvpt{30}}
\let\Huge=\huge
%</12pt>
```

(*End definition for* `\scriptsize` *and others.*)

```
%<*article|report|book>
\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
\renewcommand\baselinestretch{}
\setlength\parskip{0\p@ \@plus \p@}
%</article|report|book>
%<*10pt|11pt|12pt>
\if@twocolumn
  \setlength\parindent{1em}
\else
%<10pt>  \setlength\parindent{15\p@}
%<11pt>  \setlength\parindent{17\p@}
%<12pt>  \setlength\parindent{1.5em}
\fi
%</10pt|11pt|12pt>
%<*10pt|11pt|12pt>
\setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
\setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
```

```
\setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
%</10pt|11pt|12pt>
```

```
%<*article|report|book>
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
```

```
% \clubpenalty  150
% \widowpenalty 150
```

```
% \displaywidowpenalty 50
% \predisplaypenalty   10000
% \postdisplaypenalty  0
```

```
% \interlinepenalty 0
```

```
\brokenpenalty 100
%</article|report|book>
```

47.4.3 Page Layout

All margin dimensions are measured from a point one inch from the top and lefthand side of the page.

47.4.30 Vertical spacing

`\headheight` `\headsep` `\topskip`

The **\headheight** is the height of the box that will contain the running head. The **\headsep** is the distance between the bottom of the running head and the top of the text. The **\topskip** is the **\baselineskip** for the first line on a page; LaTeX's output routine will not work properly if it has the value 0pt, so do not do that!

```
%<*10pt|11pt|12pt>
\setlength\headheight{12\p@}
%<!bk>\setlength\headsep   {25\p@}
%<10pt&bk>\setlength\headsep   {.25in}
%<11pt&bk>\setlength\headsep   {.275in}
%<12pt&bk>\setlength\headsep   {.275in}
%<10pt>\setlength\topskip   {10\p@}
%<11pt>\setlength\topskip   {11\p@}
%<12pt>\setlength\topskip   {12\p@}
```

(*End definition for* `\headheight`, `\headsep`, *and* `\topskip`.)

```
%<!bk>\setlength\footskip{30\p@}
%<10pt&bk>\setlength\footskip{.35in}
%<11pt&bk>\setlength\footskip{.38in}
%<12pt&bk>\setlength\footskip{30\p@}
```

```
\if@compatibility \setlength\maxdepth{4\p@} \else
\setlength\maxdepth{.5\topskip} \fi
```

```
\if@compatibility
  \if@twocolumn
    \setlength\textwidth{410\p@}
```

```
  \else
%<10pt&!bk>    \setlength\textwidth{345\p@}
%<11pt&!bk>    \setlength\textwidth{360\p@}
%<12pt&!bk>    \setlength\textwidth{390\p@}
%<10pt&bk>    \setlength\textwidth{4.5in}
%<11pt&bk>    \setlength\textwidth{5in}
%<12pt&bk>    \setlength\textwidth{5in}
  \fi
\else
  \setlength\@tempdima{\paperwidth}
  \addtolength\@tempdima{-2in}
%<10pt>  \setlength\@tempdimb{345\p@}
%<11pt>  \setlength\@tempdimb{360\p@}
%<12pt>  \setlength\@tempdimb{390\p@}
  \if@twocolumn
    \ifdim\@tempdima>2\@tempdimb\relax
      \setlength\textwidth{2\@tempdimb}
    \else
      \setlength\textwidth{\@tempdima}
    \fi
  \else
    \ifdim\@tempdima>\@tempdimb\relax
      \setlength\textwidth{\@tempdimb}
    \else
      \setlength\textwidth{\@tempdima}
    \fi
  \fi
\fi
\if@compatibility\else
  \@settopoint\textwidth
\fi
\if@compatibility
%<10pt&!bk>  \setlength\textheight{43\baselineskip}
%<10pt&bk>  \setlength\textheight{41\baselineskip}
%<11pt>  \setlength\textheight{38\baselineskip}
%<12pt>  \setlength\textheight{36\baselineskip}
\else
  \setlength\@tempdima{\paperheight}
  \addtolength\@tempdima{-2in}
  \addtolength\@tempdima{-1.5in}
  \divide\@tempdima\baselineskip
  \@tempcnta=\@tempdima
  \setlength\textheight{\@tempcnta\baselineskip}
\fi
\addtolength\textheight{\topskip}
\if@twocolumn
 \setlength\marginparsep {10\p@}
```

```
\else
%<10pt&!bk>  \setlength\marginparsep{11\p@}
%<11pt&!bk>  \setlength\marginparsep{10\p@}
%<12pt&!bk>  \setlength\marginparsep{10\p@}
%<bk>  \setlength\marginparsep{7\p@}
\fi
%<10pt|11pt>\setlength\marginparpush{5\p@}
%<12pt>\setlength\marginparpush{7\p@}
```

```
\if@compatibility
%<*bk>
%<10pt>   \setlength\oddsidemargin   {.5in}
%<11pt>   \setlength\oddsidemargin   {.25in}
%<12pt>   \setlength\oddsidemargin   {.25in}
%<10pt>   \setlength\evensidemargin  {1.5in}
%<11pt>   \setlength\evensidemargin  {1.25in}
%<12pt>   \setlength\evensidemargin  {1.25in}
%<10pt>   \setlength\marginparwidth {.75in}
%<11pt>   \setlength\marginparwidth {1in}
%<12pt>   \setlength\marginparwidth {1in}
%</bk>
%<*!bk>
  \if@twoside
%<10pt>     \setlength\oddsidemargin   {44\p@}
%<11pt>     \setlength\oddsidemargin   {36\p@}
%<12pt>     \setlength\oddsidemargin   {21\p@}
%<10pt>     \setlength\evensidemargin  {82\p@}
%<11pt>     \setlength\evensidemargin  {74\p@}
%<12pt>     \setlength\evensidemargin  {59\p@}
%<10pt>     \setlength\marginparwidth {107\p@}
%<11pt>     \setlength\marginparwidth {100\p@}
%<12pt>     \setlength\marginparwidth {85\p@}
```

```
  \else
%<10pt>     \setlength\oddsidemargin   {63\p@}
%<11pt>     \setlength\oddsidemargin   {54\p@}
%<12pt>     \setlength\oddsidemargin   {39.5\p@}
%<10pt>     \setlength\evensidemargin  {63\p@}
%<11pt>     \setlength\evensidemargin  {54\p@}
%<12pt>     \setlength\evensidemargin  {39.5\p@}
%<10pt>     \setlength\marginparwidth  {90\p@}
%<11pt>     \setlength\marginparwidth  {83\p@}
%<12pt>     \setlength\marginparwidth  {68\p@}
  \fi
%</!bk>
```

```
  \if@twocolumn
     \setlength\oddsidemargin  {30\p@}
     \setlength\evensidemargin {30\p@}
     \setlength\marginparwidth {48\p@}
  \fi
```

```
\else
  \if@twoside
    \setlength\@tempdima        {\paperwidth}
    \addtolength\@tempdima      {-\textwidth}
```

```
    \setlength\oddsidemargin    {.4\@tempdima}
    \addtolength\oddsidemargin  {-1in}
    \setlength\marginparwidth   {.6\@tempdima}
    \addtolength\marginparwidth {-\marginparsep}
    \addtolength\marginparwidth {-0.4in}
  \else
    \setlength\@tempdima        {\paperwidth}
    \addtolength\@tempdima      {-\textwidth}
    \setlength\oddsidemargin    {.5\@tempdima}
    \addtolength\oddsidemargin  {-1in}
    \setlength\marginparwidth   {.5\@tempdima}
    \addtolength\marginparwidth {-\marginparsep}
    \addtolength\marginparwidth {-0.4in}
    \addtolength\marginparwidth {-.4in}
  \fi
  \ifdim \marginparwidth >2in
     \setlength\marginparwidth{2in}
  \fi
  \@settopoint\oddsidemargin
  \@settopoint\marginparwidth
  \setlength\evensidemargin  {\paperwidth}
  \addtolength\evensidemargin{-2in}
  \addtolength\evensidemargin{-\textwidth}
  \addtolength\evensidemargin{-\oddsidemargin}
  \@settopoint\evensidemargin
\fi
\if@compatibility
%<!bk>  \setlength\topmargin{27pt}
%<10pt&bk>  \setlength\topmargin{.75in}
%<11pt&bk>  \setlength\topmargin{.73in}
%<12pt&bk>  \setlength\topmargin{.73in}
\else
  \setlength\topmargin{\paperheight}
  \addtolength\topmargin{-2in}
  \addtolength\topmargin{-\headheight}
  \addtolength\topmargin{-\headsep}
  \addtolength\topmargin{-\textheight}
  \addtolength\topmargin{-\footskip}     % this might be wrong!
  \addtolength\topmargin{-.5\topmargin}
  \@settopoint\topmargin
\fi
%<10pt>\setlength\footnotesep{6.65\p@}
%<11pt>\setlength\footnotesep{7.7\p@}
%<12pt>\setlength\footnotesep{8.4\p@}
%<10pt>\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
%<11pt>\setlength{\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}
%<12pt>\setlength{\skip\footins}{10.8\p@ \@plus 4\p@ \@minus 2\p@}
%</10pt|11pt|12pt>
```

```
%<*article|report|book>
\setcounter{topnumber}{2}
\renewcommand\topfraction{.7}
\setcounter{bottomnumber}{1}
\renewcommand\bottomfraction{.3}
\setcounter{totalnumber}{3}
\renewcommand\textfraction{.2}
\renewcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\renewcommand\dbltopfraction{.7}
\renewcommand\dblfloatpagefraction{.5}
%</article|report|book>
%<*10pt>
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
%</10pt>
%<*11pt>
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep   {12\p@ \@plus 2\p@ \@minus 2\p@}
%</11pt>
%<*12pt>
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep   {14\p@ \@plus 4\p@ \@minus 4\p@}
%</12pt>
%<*10pt>
\setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
%</10pt>
%<*11pt>
\setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
%</11pt>
%<*12pt>
\setlength\dblfloatsep    {14\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
%</12pt>
%<*10pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
%</10pt>
%<*11pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
```

```
%</11pt>
%<*12pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{10\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
%</12pt>
%<*10pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
%</10pt>
%<*11pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
%</11pt>
%<*12pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{10\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
%</12pt>
%<*article|report|book>
\if@twoside
  \def\ps@headings{%
      \let\@oddfoot\@empty\let\@evenfoot\@empty
      \def\@evenhead{\thepage\hfil\slshape\leftmark}%
      \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
      \let\@mkboth\markboth
%<*article>
    \def\sectionmark##1{%
      \markboth {\MakeUppercase{%
        \ifnum \c@secnumdepth >\z@
          \thesection\quad
        \fi
        ##1}}{}}%
    \def\subsectionmark##1{%
      \markright {%
        \ifnum \c@secnumdepth >\@ne
          \thesubsection\quad
        \fi
        ##1}}}
%</article>
%<*report|book>
    \def\chaptermark##1{%
      \markboth {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
%<book>          \if@mainmatter
            \@chapapp\ \thechapter. \ %
%<book>          \fi
        \fi
        ##1}}{}}%
```

```
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\z@
          \thesection. \ %
        \fi
        ##1}}}}
%</report|book>
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
%<*article>
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \thesection\quad
        \fi
        ##1}}}}
%</article>
%<*report|book>
    \def\chaptermark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
%<book>          \if@mainmatter
            \@chapapp\ \thechapter. \ %
%<book>          \fi
        \fi
        ##1}}}}
%</report|book>
\fi
\def\ps@myheadings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\@gobbletwo
%<!article>    \let\chaptermark\@gobble
    \let\sectionmark\@gobble
%<article>    \let\subsectionmark\@gobble
    }
% \newcommand*{\title}[1]{\gdef\@title{#1}}
% \newcommand*{\author}[1]{\gdef\@author{#1}}
% \newcommand*{\date}[1]{\gdef\@date{#1}}
% \date{\today}
  \if@titlepage
  \newcommand\maketitle{\begin{titlepage}%
  \let\footnotesize\small
  \let\footnoterule\relax
  \let \footnote \thanks
```

```
  \null\vfil
  \vskip 60\p@
  \begin{center}%
    {\LARGE \@title \par}%
    \vskip 3em%
    {\large
     \lineskip .75em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
      \vskip 1.5em%
    {\large \@date \par}%       % Set date in \large size.
  \end{center}\par
  \@thanks
  \vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\else
\newcommand\maketitle{\par
  \begingroup
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
    \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
    \long\def\@makefntext##1{\parindent 1em\noindent
            \hb@xt@1.8em{%
                \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
    \if@twocolumn
      \ifnum \col@number=\@ne
        \@maketitle
      \else
        \twocolumn[\@maketitle]%
      \fi
    \else
      \newpage
      \global\@topnum\z@   % Prevents figures from going at top of page.
      \@maketitle
    \fi
    \thispagestyle{plain}\@thanks
```

```
  \endgroup
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\def\@maketitle{%
  \newpage
  \null
  \vskip 2em%
  \begin{center}%
  \let \footnote \thanks
    {\LARGE \@title \par}%
    \vskip 1.5em%
    {\large
      \lineskip .5em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
    \vskip 1em%
    {\large \@date}%
  \end{center}%
  \par
  \vskip 1.5em}
\fi
%<!article>\newcommand*\chaptermark[1]{}
% \newcommand*\sectionmark[1]{}
% \newcommand*\subsectionmark[1]{}
% \newcommand*\subsubsectionmark[1]{}
% \newcommand*\paragraphmark[1]{}
% \newcommand*\subparagraphmark[1]{}
%<article>\setcounter{secnumdepth}{3}
%<!article>\setcounter{secnumdepth}{2}
\newcounter {part}
%<article>\newcounter {section}
%<*report|book>
\newcounter {chapter}
\newcounter {section}[chapter]
%</report|book>
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
```

```
\renewcommand \thepart {\@Roman\c@part}
%<article>\renewcommand \thesection {\@arabic\c@section}
%<*report|book>
\renewcommand \thechapter {\@arabic\c@chapter}
\renewcommand \thesection {\thechapter.\@arabic\c@section}
%</report|book>
\renewcommand\thesubsection   ⤸
      ⤹{\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\renewcommand\theparagraph    ⤸
      ⤹{\thesubsubsection.\@arabic\c@paragraph}
\renewcommand\thesubparagraph ⤸
      ⤹{\theparagraph.\@arabic\c@subparagraph}
```

```
%<report|book>\newcommand\@chapapp{\chaptername}
```

```
%<*book>
\newcommand\frontmatter{%
%   \if@openright
    \cleardoublepage
%   \else
%     \clearpage
%   \fi
  \@mainmatterfalse
  \pagenumbering{roman}}
```

```
\newcommand\mainmatter{%
   \if@openright
    \cleardoublepage
   \else
     \clearpage
   \fi
  \@mainmattertrue
  \pagenumbering{arabic}}
```

```
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
%</book>
```

```
%<*article>
\newcommand\part{%
   \if@noskipsec \leavevmode \fi
   \par
   \addvspace{4ex}%
   \@afterindentfalse
   \secdef\@part\@spart}
%</article>
```

```
%<*report|book>
\newcommand\part{%
  \if@openright
    \cleardoublepage
```

```
  \else
    \clearpage
  \fi
  \thispagestyle{plain}%
```

When we are making a two column document, this will be a one column page. We use `@tempswa` to remember to switch back to two columns.

```
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
```

We need an empty box to prevent the fil glue from disappearing.

```
  \null\vfil
```

Here we use `\secdef` to indicate which commands to use to make the actual heading.

```
  \secdef\@part\@spart}
%</report|book>
```

```
%<*article>
\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
    \fi
    {\parindent \z@ \raggedright
     \interlinepenalty \@M
     \normalfont
     \ifnum \c@secnumdepth >\m@ne
       \Large\bfseries \partname\nobreakspace\thepart
       \par\nobreak
     \fi
     \huge \bfseries #2%
     \markboth{}{}\par}%
    \nobreak
    \vskip 3ex
    \@afterheading}
%</article>
```

```
%<*report|book>
\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >-2\relax
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
    \fi
    \markboth{}{}%
    {\centering
```

```
      \interlinepenalty \@M
      \normalfont
      \ifnum \c@secnumdepth >-2\relax
        \huge\bfseries \partname\nobreakspace\thepart
        \par
        \vskip 20\p@
      \fi
      \Huge \bfseries #2\par}%
    \@endpart}
%</report|book>
%<*article>
\def\@spart#1{%
    {\parindent \z@ \raggedright
      \interlinepenalty \@M
      \normalfont
      \huge \bfseries #1\par}%
      \nobreak
      \vskip 3ex
      \@afterheading}
%</article>
%<*report|book>
\def\@spart#1{%
    {\centering
      \interlinepenalty \@M
      \normalfont
      \Huge \bfseries #1\par}%
    \@endpart}
%</report|book>
%<*report|book>
\def\@endpart{\vfil\newpage
               \if@twoside
                \if@openright
                 \null
                 \thispagestyle{empty}%
                 \newpage
                \fi
               \fi
               \if@tempswa
                 \twocolumn
               \fi}
%</report|book>
%<*report|book>
\newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                    \thispagestyle{plain}%
                    \global\@topnum\z@
                    \@afterindentfalse
                    \secdef\@chapter\@schapter}
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
%<book>                        \if@mainmatter
```

```
                          \refstepcounter{chapter}%
                          \typeout{\@chapapp\space\thechapter.}%
                          \addcontentsline{toc}{chapter}%
                                    {\protect\numberline{\thechapter}#1}%
%<*book>
                        \else
                          \addcontentsline{toc}{chapter}{#1}%
                        \fi
%</book>
                      \else
                        \addcontentsline{toc}{chapter}{#1}%
                      \fi
                      \chaptermark{#1}%
                      \addtocontents{lof}{\protect\addvspace{10\p@}}%
                      \addtocontents{lot}{\protect\addvspace{10\p@}}%
                      \if@twocolumn
                        \@topnewpage[\@makechapterhead{#2}]%
                      \else
                        \@makechapterhead{#2}%
                        \@afterheading
                      \fi}
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
%<book>      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
%<book>      \fi
    \fi
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
    \vskip 40\p@
  }}
\def\@schapter#1{\if@twocolumn
                    \@topnewpage[\@makeschapterhead{#1}]%
                  \else
                    \@makeschapterhead{#1}%
                    \@afterheading
                  \fi}
\def\@makeschapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright
    \normalfont
    \interlinepenalty\@M
    \Huge \bfseries  #1\par\nobreak
    \vskip 40\p@
  }}
%</report|book>
```

```
\newcommand\section{\@startsection {section}{1}{\z@}%
                                    {-3.5ex \@plus -1ex ↩
                                          ↪\@minus -.2ex}%
                                    {2.3ex \@plus.2ex}%
                                    {\normalfont\Large\bfseries}}
\newcommand\subsection{\@startsection{subsection}{2}{\z@}%
                                      {-3.25ex\@plus -1ex ↩
                                            ↪\@minus -.2ex}%
                                      {1.5ex \@plus .2ex}%
                                      {\normalfont\large\bfseries}}
\newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                      {-3.25ex\@plus -1ex ↩
                                            ↪\@minus -.2ex}%
                                      {1.5ex \@plus .2ex}%
                                      {\normalfont\normalsize\bfseries}}
\newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
                                     {3.25ex \@plus1ex ↩
                                           ↪\@minus.2ex}%
                                     {-1em}%
                                     {\normalfont\normalsize\bfseries}}
\newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                        {3.25ex \@plus1ex ↩
                                              ↪\@minus .2ex}%
                                        {-1em}%
                                        {\normalfont\normalsize\bfseries}}
```

47.4.4 Lists

47.4.40 General List Parameters

The following commands are used to set the default values for the list environment's parameters. See the LaTeX manual for an explanation of the meanings of the parameters. Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, for a Kth level list, the command `\@listK` is called, where 'K' denotes 'i', 'i', ... , 'vi'. (I.e., `\@listiii` is called for a third-level list.) By convention, `\@listK` should set `\leftmargin` to `\leftmarginK`. When we are in two column mode some of the margins are set somewhat smaller.

```
\if@twocolumn
  \setlength\leftmargini  {2em}
\else
  \setlength\leftmargini  {2.5em}
\fi
\leftmargin  \leftmargini
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv  {1.7em}
\if@twocolumn
  \setlength\leftmarginv  {.5em}
```

```
  \setlength\leftmarginvi {.5em}
\else
  \setlength\leftmarginv  {1em}
  \setlength\leftmarginvi {1em}
\fi
\setlength  \labelsep  {.5em}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
%</article|report|book>
%<10pt>\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
%<11pt>\setlength\partopsep{3\p@ \@plus 1\p@ \@minus 1\p@}
%<12pt>\setlength\partopsep{3\p@ \@plus 2\p@ \@minus 2\p@}
%<*article|report|book>
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
%</article|report|book>
%<*10pt|11pt|12pt>
\def\@listi{\leftmargin\leftmargini
%<*10pt>
            \parsep 4\p@ \@plus2\p@ \@minus\p@
            \topsep 8\p@ \@plus2\p@ \@minus4\p@
            \itemsep4\p@ \@plus2\p@ \@minus\p@}
%</10pt>
%<*11pt>
            \parsep 4.5\p@ \@plus2\p@ \@minus\p@
            \topsep 9\p@   \@plus3\p@ \@minus5\p@
            \itemsep4.5\p@ \@plus2\p@ \@minus\p@}
%</11pt>
%<*12pt>
            \parsep 5\p@  \@plus2.5\p@ \@minus\p@
            \topsep 10\p@ \@plus4\p@   \@minus6\p@
            \itemsep5\p@  \@plus2.5\p@ \@minus\p@}
%</12pt>
\let\@listI\@listi
\@listi
\def\@listii {\leftmargin\leftmarginii
              \labelwidth\leftmarginii
              \advance\labelwidth-\labelsep
%<*10pt>
              \topsep    4\p@ \@plus2\p@ \@minus\p@
              \parsep    2\p@ \@plus\p@  \@minus\p@
%</10pt>
%<*11pt>
              \topsep    4.5\p@ \@plus2\p@ \@minus\p@
              \parsep    2\p@   \@plus\p@  \@minus\p@
%</11pt>
%<*12pt>
              \topsep    5\p@   \@plus2.5\p@ \@minus\p@
              \parsep    2.5\p@ \@plus\p@    \@minus\p@
%</12pt>
```

```
                \itemsep   \parsep}
\def\@listiii{\leftmargin\leftmargininiii
                \labelwidth\leftmargininiii
                \advance\labelwidth-\labelsep
%<10pt>               \topsep    2\p@ \@plus\p@\@minus\p@
%<11pt>               \topsep    2\p@ \@plus\p@\@minus\p@
%<12pt>               \topsep    2.5\p@\@plus\p@\@minus\p@
                \parsep    \z@
                \partopsep \p@ \@plus\z@ \@minus\p@
                \itemsep   \topsep}
\def\@listiv {\leftmargin\leftmargininiv
                \labelwidth\leftmargininiv
                \advance\labelwidth-\labelsep}
\def\@listv  {\leftmargin\leftmarginv
                \labelwidth\leftmarginv
                \advance\labelwidth-\labelsep}
\def\@listvi {\leftmargin\leftmarginvi
                \labelwidth\leftmarginvi
                \advance\labelwidth-\labelsep}
%</10pt|11pt|12pt>
```

```
%<*article|report|book>
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
```

```
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
```

```
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
```

47.4.40 Itemize

`\labelitemi`
`\labelitemii`
`\labelitemiii`
`\labelitemiv`

Itemization is controlled by four commands: **\labelitemi**, **\labelitemii**, **\labelitemiii**, and **\labelitemiv**, which define the labels of thevarious itemization levels: the symbols used are bullet, bold en-dash, centered asterisk and centred dot.

```
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}
```

(*End definition for* `\labelitemi` *and others.*)

47.4.40 Description

```
\newenvironment{description}
                {\list{}{\labelwidth\z@ \itemindent-\leftmargin
```

```
                        \let\makelabel\descriptionlabel}}
               {\endlist}
\newcommand*\descriptionlabel[1]{\hspace\labelsep
                                 \normalfont\bfseries #1}
% \changes{v1.3m}{1995/10/23}{Added setting of \cs{beginparpenalty} to
%    discourage page break before abstract heading.}
%<*article|report>
\if@titlepage
  \newenvironment{abstract}{%
      \titlepage
      \null\vfil
      \@beginparpenalty\@lowpenalty
      \begin{center}%
        \bfseries \abstractname
        \@endparpenalty\@M
      \end{center}}%
     {\par\vfil\null\endtitlepage}
\else
  \newenvironment{abstract}{%
      \if@twocolumn
        \section*{\abstractname}%
      \else
        \small
        \begin{center}%
          {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
        \end{center}%
        \quotation
      \fi}
      {\if@twocolumn\else\endquotation\fi}
\fi
%</article|report>
\newenvironment{verse}
               {\let\\\@centercr
                \list{}{\itemsep      \z@
                        \itemindent   -1.5em%
                        \listparindent\itemindent
                        \rightmargin  \leftmargin
                        \advance\leftmargin 1.5em}%
                \item\relax}
               {\endlist}
\newenvironment{quotation}
               {\list{}{\listparindent 1.5em%
                        \itemindent    \listparindent
                        \rightmargin   \leftmargin
                        \parsep        \z@ \@plus\p@}%
                \item\relax}
               {\endlist}
\newenvironment{quote}
               {\list{}{\rightmargin\leftmargin}%
                \item\relax}
               {\endlist}
```

```
\if@compatibility
\newenvironment{titlepage}
    {%
%<book>      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\z@
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
    }
\else
\newenvironment{titlepage}
    {%
%<book>      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\@ne
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
     \if@twoside\else
        \setcounter{page}\@ne
     \fi
    }
\fi
%<*article>
\newcommand\appendix{\par
  \setcounter{section}{0}%
  \setcounter{subsection}{0}%
  \gdef\thesection{\@Alph\c@section}}
%</article>
%<*report|book>
\newcommand\appendix{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}%
  \gdef\thechapter{\@Alph\c@chapter}}
%</report|book>
\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}
\setlength\tabbingsep{\labelsep}
```

```
\skip\@mpfootins = \skip\footins
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
%<article>\renewcommand \theequation {\@arabic\c@equation}
%<*report|book>
\@addtoreset {equation}{chapter}
\renewcommand\theequation
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
%</report|book>
% \setlength\jot{3pt}
% \def\@eqnnum{(\theequation)}
%<*article>
\newcounter{figure}
\renewcommand \thefigure {\@arabic\c@figure}
%</article>
%<*report|book>
\newcounter{figure}[chapter]
\renewcommand \thefigure
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
%</report|book>
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
%<*article>
\newcounter{table}
\renewcommand\thetable{\@arabic\c@table}
%</article>
%<*report|book>
\newcounter{table}[chapter]
\renewcommand \thetable
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
%</report|book>
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\nobreakspace\thetable}
\newenvironment{table}
               {\@float{table}}
               {\end@float}
\newenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{10\p@}
\setlength\belowcaptionskip{0\p@}
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1: #2}%
  \ifdim \wd\@tempboxa >\hsize
    #1: #2\par
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
\newcommand\@pnumwidth{1.55em}
\newcommand\@tocrmarg{2.55em}
\newcommand\@dotsep{4.5}
%<article>\setcounter{tocdepth}{3}
%<!article>\setcounter{tocdepth}{2}
\newcommand\tableofcontents{%
%<*report|book>
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\contentsname
%</report|book>
%<article>    \section*{\contentsname
        \@mkboth{%
           \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%
    \@starttoc{toc}%
%<!article>    \if@restonecol\twocolumn\fi
    }
\newcommand*\l@part[2]{%
  \ifnum \c@tocdepth >-2\relax
%<article>    \addpenalty\@secpenalty
%<!article>    \addpenalty{-\@highpenalty}%
    \addvspace{2.25em \@plus\p@}%
```

```
    \setlength\@tempdima{3em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      {\leavevmode
       \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
       \nobreak
%<article>       \if@compatibility
         \global\@nobreaktrue
         \everypar{\global\@nobreakfalse\everypar{}}%
%<article>      \fi
    \endgroup
  \fi}
%<*report|book>
\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip 1.0em \@plus\p@
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
%</report|book>
%<*article>
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
    \endgroup
  \fi}
%</article>
```

```
%<*report|book>
\newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
%</report|book>
```

```
%<*article>
\newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
%</article>
%<*report|book>
\newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
\newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
%</report|book>
```

```
\newcommand\listoffigures{%
%<*report|book>
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listfigurename}%
%</report|book>
%<article>    \section*{\listfigurename}%
      \@mkboth{\MakeUppercase\listfigurename}%
               {\MakeUppercase\listfigurename}%
    \@starttoc{lof}%
%<report|book>    \if@restonecol\twocolumn\fi
    }
```

```
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
```

```
\newcommand\listoftables{%
%<*report|book>
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listtablename}%
%</report|book>
%<article>    \section*{\listtablename}%
      \@mkboth{%
          \MakeUppercase\listtablename}%
         {\MakeUppercase\listtablename}%
    \@starttoc{lot}%
%<report|book>    \if@restonecol\twocolumn\fi
    }
```

```
\let\l@table\l@figure
```

```
\newdimen\bibindent
\setlength\bibindent{1.5em}
```

```
\newenvironment{thebibliography}[1]
%<*article>
     {\section*{\refname}%
      \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
%</article>
%<*!article>
     {\chapter*{\bibname}%
      \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
%</!article>
      \list{\@biblabel{\@arabic\c@enumiv}}%
           {\settowidth\labelwidth{\@biblabel{#1}}%
            \leftmargin\labelwidth
            \advance\leftmargin\labelsep
            \@openbib@code
            \usecounter{enumiv}%
            \let\p@enumiv\@empty
            \renewcommand\theenumiv{\@arabic\c@enumiv}}%
      \sloppy
      \clubpenalty4000
      \@clubpenalty \clubpenalty
      \widowpenalty4000%
      \sfcode`\.\@m}
     {\def\@noitemerr
       {\@latex@warning{Empty `thebibliography' environment}}%
      \endlist}
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
\let\@openbib@code\@empty
% \renewcommand*{\@biblabel}[1]{[#1]\hfill}
% \renewcommand*{\@cite}[1]{[#1]}
\newenvironment{theindex}
               {\if@twocolumn
                  \@restonecolfalse
                \else
                  \@restonecoltrue
                \fi
%<article>                \twocolumn[\section*{\indexname}]%
%<!article>                \twocolumn[\@makeschapterhead{\indexname}]%
                \@mkboth{\MakeUppercase\indexname}%
                        {\MakeUppercase\indexname}%
                \thispagestyle{plain}\parindent\z@
                \parskip\z@ \@plus .3\p@\relax
                \columnseprule \z@
                \columnsep 35\p@
                \let\item\@idxitem}
               {\if@restonecol\onecolumn\else\clearpage\fi}
\newcommand\@idxitem{\par\hangindent 40\p@}
\newcommand\subitem{\@idxitem \hspace*{20\p@}}
\newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
```

```
\newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}
%<!article>\@addtoreset{footnote}{chapter}
\newcommand\@makefntext[1]{%
    \parindent 1em%
    \noindent
    \hb@xt@1.8em{\hss\@makefnmark}#1}
%\renewcommand\@makefnmark{\hbox{\@textsuperscript
%                                  {\normalfont\@thefnmark}}}
\newcommand\contentsname{Contents}
\newcommand\listfigurename{List of Figures}
\newcommand\listtablename{List of Tables}
%<article>\newcommand\refname{References}
%<report|book>\newcommand\bibname{Bibliography}
\newcommand\indexname{Index}
\newcommand\figurename{Figure}
\newcommand\tablename{Table}
\newcommand\partname{Part}
%<report|book>\newcommand\chaptername{Chapter}
\newcommand\appendixname{Appendix}
%<!book>\newcommand\abstractname{Abstract}
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
\setlength\columnsep{10\p@}
\setlength\columnseprule{0\p@}
%<!book>\pagestyle{plain}
%<book>\pagestyle{headings}
\pagenumbering{arabic}
\if@twoside
\else
  \raggedbottom
\fi
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
\fi
%</article|report|book>
```

Book Design

phd package and give a background as to how fonts are used in LaTeX.

phd package and give a background as to how fonts are used in LaTeX.

48.1 First Steps

In this chapter we will develop a full book template from scratch. Before we delve into it further, I would like to emphasize that the **phd** system is a bit different from classes. A **phd** style includes all the information necessary for the typesetting of a document. I have called this a style template. It is slightly different from a class system where generic commands might be included that can develop a totally different look. An identical design with perhaps different colors and fonts and other minor changes, is termed a *theme*.

Unlike book designers who would first focus on fonts, we will first give our attention to the structural elements of the book. I will be using as an example the *Linear Algebra*.

xvi A Note to Students

Study Guide

To help you succeed in this course, I suggest that you purchase the *Study Guide* (*www.mypearsonstore.com*; 0-321-38883-6). Not only will it help you learn linear algebra, it also will show you how to study mathematics. At strategic points in your textbook, an icon [ss] will direct you to special subsections in the *Study Guide* entitled "Mastering Linear Algebra Concepts." There you will find suggestions for constructing effective review sheets of key concepts. The act of preparing the sheets is one of the secrets to success in the course, because you will construct *links between ideas*. These links are the "glue" that enables you to build a solid foundation for learning and *remembering* the main concepts in the course.

The *Study Guide* contains a detailed solution to every third odd-numbered exercise, plus solutions to all odd-numbered writing exercises for which only a hint is given in the *Answers* section of this book. The *Guide* is separate from the text because you must learn to write solutions by yourself, without much help. (I know from years of experience that easy access to solutions in the back of the text slows the mathematical development of most students.) The *Guide* also provides warnings of common errors and helpful hints that call attention to key exercises and potential exam questions.

If you have access to technology—MATLAB, Maple, Mathematica, or a TI graphing calculator—you can save many hours of homework time. The *Study Guide* is your "lab manual" that explains how to use each of these matrix utilities. It introduces new commands when they are needed. You can download from the website *www.pearsonhighered.com/lay* the data for more than 850 exercises in the text. (With a few keystrokes, you can display any numerical homework problem on your screen.) Special matrix commands will perform the computations for you!

What you do in your first few weeks of studying this course will set your pattern for the term and determine how well you finish the course. Please read "How to Study Linear Algebra" in the *Study Guide* as soon as possible. My students have found the strategies there very helpful, and I hope you will, too.

1 Linear Equations in Linear Algebra

INTRODUCTORY EXAMPLE

Linear Models in Economics and Engineering

It was late summer in 1949. Harvard Professor Wassily Leontief was carefully feeding the last of his punched cards into the university's Mark II computer. The cards contained economic information about the U.S. economy and represented a summary of more than 250,000 pieces of information produced by the U.S. Bureau of Labor Statistics after two years of intensive work. Leontief had divided the U.S. economy into 500 "sectors," such as the coal industry, the automotive industry, communications, and so on. For each sector, he had written a linear equation that described how the sector distributed its output to the other sectors of the economy. Because the Mark II, one of the largest computers of its day, could not handle the resulting system of 500 equations in 500 unknowns, Leontief had distilled the problem into a system of 42 equations in 42 unknowns.

Programming the Mark II computer for Leontief's 42 equations had required several months of effort, and he was anxious to see how long the computer would take to solve the problem. The Mark II hummed and blinked for 56 hours before finally producing a solution. We will discuss the nature of this solution in Sections 1.6 and 2.6.

Leontief, who was awarded the 1973 Nobel Prize in Economic Science, opened the door to a new era in mathematical modeling in economics. His efforts at Harvard in 1949 marked one of the first significant uses of computers to analyze what was then a large-scale mathematical model. Since that time, researchers in many other fields have employed computers to analyze mathematical models. Because of the massive amounts of data involved, the models are usually *linear*; that is, they are described by *systems of linear equations*.

The importance of linear algebra for applications has risen in direct proportion to the increase in computing power, with each new generation of hardware and software triggering a demand for even greater capabilities. Computer science is thus intricately linked with linear algebra through the explosive growth of parallel processing and large-scale computations.

Scientists and engineers now work on problems far more complex than even dreamed possible a few decades ago. Today, linear algebra has more potential value for students in many scientific and business fields than any other undergraduate mathematics subject! The material in this text provides the foundation for further work in many interesting areas. Here are a few possibilities; others will be described later.

• *Oil exploration.* When a ship searches for offshore oil deposits, its computers solve thousands of separate systems of linear equations *every day*. The

1

7.75 x 10.24 in

Figure 48.1: The opening chapter can leave a blank page.

Figure **??** shows the chapter head design. This is an interesting and challenging design that we will not easily make with the **phd** standard chapter head routines. The chapter starts with a full line and a structural element that is called *Introductory Example*. The heading of this also goes to the Table of Contents. So the chapter opening page starts with a rule and end with a rule. The ending rule in Figure **??** can be seen in the next figure.

48.1.1 Chapter Opening

One of the first things you will need to take care of, is to design if the style template should cater for opening at right or if it is to open at any place. Another decision you will need to make, is what to do with blank pages. Personally I dislike them and suggest, if you are going to have them to either introduce epigraphs or full page images.

This page intentionally left blank

3 Determinants

WEB

INTRODUCTORY EXAMPLE

Random Paths and Distortion

In his autobiographical book "Surely You're Joking, Mr. Feynman," the Nobel Prize–winning physicist Richard Feynman tells of observing ants in his Princeton graduate school apartment. He studied the ants' behavior by providing paper ferries to sugar suspended on a string where the ants would not accidentally find it. When an ant would step onto a paper ferry, Feynman would transport the ant to the food and then back. After the ants learned to use the ferry, he relocated the return landing. The colony soon confused the outbound and return ferry landings, indicating that their "learning" consisted of creating and following trails. Feynman confirmed this conjecture by laying glass slides on the floor. Once the ants established trails on the glass slides, he rearranged the slides and therefore the trails on them. The ants followed the repositioned trails and Feynman could direct the ants where he wished.

Suppose Feynman had decided to conduct additional investigations using a globe built of wire mesh on which an ant must follow individual wires and choose between going left and right at each intersection. If several ants and an equal number of food sources are placed on the globe, how likely is it that each ant would find its own food source rather than encountering another ant's trail and following it to a shared resource?[5]

[1] The solution to the ant-path problem (and two other applications) can be found in a June 2005, *Mathematical Monthly* article by Arthur Benjamin and Naomi Cameron.

In order to record the actual routes of the ants and to communicate the results to others, it is convenient to use a rectangular map of the globe. There are many ways to create such maps. One simple way is to use the longitude and latitude on the globe as x and y coordinates on the map. As is the case with all maps, the result is not a faithful representation of the globe. Features near the "equator" look much the same on the globe and the map, but regions near the "poles" of the globe are distorted. Images of polar regions are much larger than the images of similar sized regions near the equator. To fit in with its surroundings on the map, the image of an ant near one of the poles should be larger than one near the equator. How much larger?

Surprisingly, both the ant-path and the area distortion problems are best answered through the use of the determinant, the subject of this chapter. Indeed, the determinant has so many uses that a summary of the applications known in the early 1900's filled a four volume treatise by Thomas Muir. With changes in emphasis and the greatly increased sizes of the matrices used in modern applications, many uses that were important then are no longer critical today. Nevertheless, the determinant still plays an important role.

163

Figure 48.2: The opening chapter can leave a blank page.

48.1.2 The User Commands

It is always best to start thinking about the user commands, as we go along in order to provide a user friendly interface, without the introduction of too many keys. We also need to name our template. We will name it *andrea* in honour of the Designer of the book, who was Andrea Nix. Andrea designed many of the Pearson books that were mostly textbooks and has a unique distinctive design style that can make a mathematics book fun to read.

```
\cxset{chapter template = andrea,
          chapter opening = right}
```

Final page layout dimensions and booleans \paperwidth 597.50787pt
\paperheight 845.04684pt
\textwidth 345.0pt
\textheight 598.0pt
\oddsidemargin 53.0pt
\evensidemargin 54.0pt
\topmargin 23.0pt
\headheight 12.0pt
\headsep 18.06749pt
\footskip 25.29494pt
\marginparwidth 96.0pt
\marginparsep 7.0pt
\columnsep 20.0pt
\columnseprule 0.0pt
\skip\footins 253

\hoffset 0.opt
\voffset 0.opt
\mag 1000

\@reversemargintrue
As we will not be sure our calculations are right or wrong (the rules can disappear at the edge of the page) I have taken 5pt out from the left or right parameters to see that we have done the calculations properly.
We also need to check on oddside pages as well. Remember the switch `\@mparswitchfalse` will set the margin pars to be on the same size. This layout only has them on the right pages. We need to set it to false.
Another decision we need to make is if we going to draw the layout using TeX commands or one of the graphic units. Using TikZ, can be much easier, but we need to ensure we know where we are on the page. Alternatively we can use the remember picture, overlay hack to accomplish it. We will first give it a try with rules and boxes.
Now we have the dimensions of the left margin and right margin width right we can continue with the layout.
The next item we will draw is the corner frame.

45.1 Sections

48.2 Sections

The sections follow a very similar style to that of the chapter heading with rules and similar colours.

2 CHAPTER 1 Linear Equations in Linear Algebra

seismic data for the equations are obtained from underwater shock waves created by explosions from air guns. The waves bounce off subsurface rocks and are measured by geophones attached to mile-long cables behind the ship.

- *Linear programming*. Many important management decisions today are made on the basis of linear programming models that utilize hundreds of variables. The airline industry, for instance, employs linear programs that schedule flight crews, monitor the locations of aircraft, or plan the varied schedules of support services such as maintenance and terminal operations.
- *Electrical networks*. Engineers use simulation software to design electrical circuits and microchips involving millions of transistors. Such software relies on linear algebra techniques and systems of linear equations.

WEB

Systems of linear equations lie at the heart of linear algebra, and this chapter uses them to introduce some of the central concepts of linear algebra in a simple and concrete setting. Sections 1.1 and 1.2 present a systematic method for solving systems of linear equations. This algorithm will be used for computations throughout the text. Sections 1.3 and 1.4 show how a system of linear equations is equivalent to a *vector equation* and to a *matrix equation*. This equivalence will reduce problems involving linear combinations of vectors to questions about systems of linear equations. The fundamental concepts of spanning, linear independence, and linear transformations, studied in the second half of the chapter, will play an essential role throughout the text as we explore the beauty and power of linear algebra.

1.1 SYSTEMS OF LINEAR EQUATIONS

A **linear equation** in the variables $x_1, \ldots, x_n$ is an equation that can be written in the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \tag{1}$$

where b and the **coefficients** $a_1, \ldots, a_n$ are real or complex numbers, usually known in advance. The subscript n may be any positive integer. In textbook examples and exercises, n is normally between 2 and 5. In real-life problems, n might be 50 or 5000, or even larger.

The equations

$$4x_1 - 5x_2 + 2 = x_1 \quad \text{and} \quad x_2 = 2(\sqrt{6} - x_1) + x_3$$

are both linear because they can be rearranged algebraically as in equation (1):

$$3x_1 - 5x_2 = -2 \quad \text{and} \quad 2x_1 + x_2 - x_3 = 2\sqrt{6}$$

The equations

$$4x_1 - 5x_2 = x_1x_2 \quad \text{and} \quad x_2 = 2\sqrt{x_1} - 6$$

are not linear because of the presence of x_1x_2 in the first equation and $\sqrt{x_1}$ in the second.

A **system of linear equations** (or a **linear system**) is a collection of one or more linear equations involving the same variables—say, $x_1, \ldots, x_n$. An example is

$$\begin{aligned} 2x_1 - x_2 + 1.5x_3 &= 8 \\ x_1 \quad - 4x_3 &= -7 \end{aligned} \tag{2}$$

1.1 Systems of Linear Equations 3

A **solution** of the system is a list $(s_1, s_2, \ldots, s_n)$ of numbers that makes each equation a true statement when the values $s_1, \ldots, s_n$ are substituted for $x_1, \ldots, x_n$, respectively. For instance, $(5, 6.5, 3)$ is a solution of system (2) because, when these values are substituted in (2) for x_1, x_2, x_3, respectively, the equations simplify to $8 = 8$ and $-7 = -7$.

The set of all possible solutions is called the **solution set** of the linear system. Two linear systems are called **equivalent** if they have the same solution set. That is, each solution of the first system is a solution of the second system, and each solution of the second system is a solution of the first.

Finding the solution set of a system of two linear equations in two variables is easy because it amounts to finding the intersection of two lines. A typical problem is

$$\begin{aligned} x_1 - 2x_2 &= -1 \\ -x_1 + 3x_2 &= \ \ 3 \end{aligned}$$

The graphs of these equations are lines, which we denote by ℓ_1 and ℓ_2. A pair of numbers (x_1, x_2) satisfies *both* equations in the system if and only if the point (x_1, x_2) lies on both ℓ_1 and ℓ_2. In the system above, the solution is the single point $(3, 2)$, as you can easily verify. See Fig. 1.

FIGURE 1 Exactly one solution.

Of course, two lines need not intersect in a single point—they could be parallel, or they could coincide and hence "intersect" at every point on the line. Figure 2 shows the graphs that correspond to the following systems:

$$\text{(a)} \quad \begin{aligned} x_1 - 2x_2 &= -1 \\ -x_1 + 2x_2 &= \ \ 3 \end{aligned} \qquad \text{(b)} \quad \begin{aligned} x_1 - 2x_2 &= -1 \\ -x_1 + 2x_2 &= \ \ 1 \end{aligned}$$

FIGURE 2 (a) No solution. (b) Infinitely many solutions.

Figures 1 and 2 illustrate the following general fact about linear systems, to be verified in Section 1.2.

Figure 48.3: The opening chapter can leave a blank page.

The book does not use subsection. As a matter of fact most books don't consider that numbering of subsections offers an advantage to the reader.

36 CHAPTER 1 Linear Equations in Linear Algebra

THEOREM 3 If A is an $m \times n$ matrix, with columns $\mathbf{a}_1, \ldots, \mathbf{a}_n$, and if $\mathbf{b}$ is in $\mathbb{R}^m$, the matrix equation

$$A\mathbf{x} = \mathbf{b} \tag{4}$$

has the same solution set as the vector equation

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_n\mathbf{a}_n = \mathbf{b} \tag{5}$$

which, in turn, has the same solution set as the system of linear equations whose augmented matrix is

$$\begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n & \mathbf{b} \end{bmatrix} \tag{6}$$

Theorem 3 provides a powerful tool for gaining insight into problems in linear algebra, because a system of linear equations may now be viewed in three different but equivalent ways: as a matrix equation, as a vector equation, or as a system of linear equations. Whenever you construct a mathematical model of a problem in real life, you are free to choose whichever viewpoint is most natural. Then you may switch from one formulation of a problem to another whenever it is convenient. In any case, the matrix equation (4), the vector equation (5), and the system of equations are all solved in the same way—by row reducing the augmented matrix (6). Other methods of solution will be discussed later.

Existence of Solutions

The definition of $A\mathbf{x}$ leads directly to the following useful fact.

The equation $A\mathbf{x} = \mathbf{b}$ has a solution if and only if $\mathbf{b}$ is a linear combination of the columns of A.

Section 1.3 considered the existence question, "Is $\mathbf{b}$ in Span $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$?" Equivalently, "Is $A\mathbf{x} = \mathbf{b}$ consistent?" A harder existence problem is to determine whether the equation $A\mathbf{x} = \mathbf{b}$ is consistent *for all* possible $\mathbf{b}$.

EXAMPLE 3 Let $A = \begin{bmatrix} 1 & 3 & 4 \\ -4 & 2 & -6 \\ -3 & -2 & -7 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$. Is the equation $A\mathbf{x} = \mathbf{b}$ consistent for all possible b_1, b_2, b_3?

SOLUTION Row reduce the augmented matrix for $A\mathbf{x} = \mathbf{b}$:

$$\begin{bmatrix} 1 & 3 & 4 & b_1 \\ -4 & 2 & -6 & b_2 \\ -3 & -2 & -7 & b_3 \end{bmatrix} \sim \begin{bmatrix} 1 & 3 & 4 & b_1 \\ 0 & 14 & 10 & b_2 + 4b_1 \\ 0 & 7 & 5 & b_3 + 3b_1 \end{bmatrix}$$
$$\sim \begin{bmatrix} 1 & 3 & 4 & b_1 \\ 0 & 14 & 10 & b_2 + 4b_1 \\ 0 & 0 & 0 & b_3 + 3b_1 - \frac{1}{2}(b_2 + 4b_1) \end{bmatrix}$$

The third entry in column 4 equals $b_1 - \frac{1}{2}b_2 + b_3$. The equation $A\mathbf{x} = \mathbf{b}$ is *not* consistent for every $\mathbf{b}$ because some choices of $\mathbf{b}$ can make $b_1 - \frac{1}{2}b_2 + b_3$ nonzero. ■

1.4 The Matrix Equation $A\mathbf{x} = \mathbf{b}$ 37

FIGURE 1 The columns of $A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix}$ span a plane through 0.

The reduced matrix in Example 3 provides a description of all $\mathbf{b}$ for which the equation $A\mathbf{x} = \mathbf{b}$ *is* consistent: The entries in $\mathbf{b}$ must satisfy

$$b_1 - \tfrac{1}{2}b_2 + b_3 = 0$$

This is the equation of a plane through the origin in $\mathbb{R}^3$. The plane is the set of all linear combinations of the three columns of A. See Fig. 1.

The equation $A\mathbf{x} = \mathbf{b}$ in Example 3 fails to be consistent for all $\mathbf{b}$ because the echelon form of A has a row of zeros. If A had a pivot in all three rows, we would not care about the calculations in the augmented column because in this case an echelon form of the augmented matrix could not have a row such as $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$.

In the next theorem, the sentence "The columns of A span $\mathbb{R}^m$" means that *every* $\mathbf{b}$ in $\mathbb{R}^m$ is a linear combination of the columns of A. In general, a set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ in $\mathbb{R}^m$ **spans** (or **generates**) $\mathbb{R}^m$ if every vector in $\mathbb{R}^m$ is a linear combination of $\mathbf{v}_1, \ldots, \mathbf{v}_p$—that is, if Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\} = \mathbb{R}^m$.

THEOREM 4 Let A be an $m \times n$ matrix. Then the following statements are logically equivalent. That is, for a particular A, either they are all true statements or they are all false.

a. For each $\mathbf{b}$ in $\mathbb{R}^m$, the equation $A\mathbf{x} = \mathbf{b}$ has a solution.
b. Each $\mathbf{b}$ in $\mathbb{R}^m$ is a linear combination of the columns of A.
c. The columns of A span $\mathbb{R}^m$.
d. A has a pivot position in every row.

Theorem 4 is one of the most useful theorems in this chapter. Statements (a), (b), and (c) are equivalent because of the definition of $A\mathbf{x}$ and what it means for a set of vectors to span $\mathbb{R}^m$. The discussion after Example 3 suggests why (a) and (d) are equivalent; a proof is given at the end of the section. The exercises will provide examples of how Theorem 4 is used.

Warning: Theorem 4 is about a *coefficient matrix*, not an augmented matrix. If an augmented matrix $\begin{bmatrix} A & \mathbf{b} \end{bmatrix}$ has a pivot position in every row, then the equation $A\mathbf{x} = \mathbf{b}$ may or may not be consistent.

Computation of $A\mathbf{x}$

The calculations in Example 1 were based on the definition of the product of a matrix A and a vector $\mathbf{x}$. The following simple example will lead to a more efficient method for calculating the entries in $A\mathbf{x}$ when working problems by hand.

EXAMPLE 4 Compute $A\mathbf{x}$, where $A = \begin{bmatrix} 2 & 3 & 4 \\ -1 & 5 & -3 \\ 6 & -2 & 8 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$.

Figure 48.4: The opening chapter can leave a blank page.

48.3 Examples and Solutions

The examples are straight forward typesetting and numbering. The specification should be that they be numbered consequently with the example and solution in capital letters to be distinguished by the type size. The colour is to be identical. The example heading is inlined with about a quadd of space between it and the text that follow. The solution is on its own line and it is followed normally by a list which is numbered alphabetically. In other cases it is in-lined see the page at the left. We can perhaps handle this with a starred command, one for stand alone heading and another for an inlined. I will come back with some suggestions for this before, we delve into codin.

48.4 Exercises

These are modelled after sections and are also numbered. They are numbered in a different counter from that of sections and are reset at every chapter.

48.5 Figures and diagrams

The user commands should also be minimized and would follow normal LaTeX conventions, with the exception we will redefine an environment `\begin`⟨*marginfigure*⟩.... The margin figures are both numbered as well as

58 CHAPTER 1 Linear Equations in Linear Algebra

The arguments in Example 3 show that you can always decide by *inspection* when a set of two vectors is linearly dependent. Row operations are unnecessary. Simply check whether at least one of the vectors is a scalar times the other. (The test applies only to sets of *two* vectors.)

Linearly dependent

Linearly independent

FIGURE 1

A set of two vectors $\{\mathbf{v}_1, \mathbf{v}_2\}$ is linearly dependent if at least one of the vectors is a multiple of the other. The set is linearly independent if and only if neither of the vectors is a multiple of the other.

In geometric terms, two vectors are linearly dependent if and only if they lie on the same line through the origin. Figure 1 shows the vectors from Example 3.

Sets of Two or More Vectors

The proof of the next theorem is similar to the solution of Example 3. Details are given at the end of this section.

THEOREM 7

Characterization of Linearly Dependent Sets

An indexed set $S = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ of two or more vectors is linearly dependent if and only if at least one of the vectors in S is a linear combination of the others. In fact, if S is linearly dependent and $\mathbf{v}_1 \neq \mathbf{0}$, then some $\mathbf{v}_j$ (with $j > 1$) is a linear combination of the preceding vectors, $\mathbf{v}_1, \dots, \mathbf{v}_{j-1}$.

Warning: Theorem 7 does *not* say that *every* vector in a linearly dependent set is a linear combination of the preceding vectors. A vector in a linearly dependent set may fail to be a linear combination of the other vectors. See Practice Problem 3.

EXAMPLE 4 Let $\mathbf{u} = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1 \\ 6 \\ 0 \end{bmatrix}$. Describe the set spanned by $\mathbf{u}$ and $\mathbf{v}$, and explain why a vector $\mathbf{w}$ is in Span $\{\mathbf{u}, \mathbf{v}\}$ if and only if $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is linearly dependent.

SOLUTION The vectors $\mathbf{u}$ and $\mathbf{v}$ are linearly independent because neither vector is a multiple of the other, and so they span a plane in $\mathbb{R}^3$. (See Section 1.3.) In fact, Span $\{\mathbf{u}, \mathbf{v}\}$ is the x_1x_2-plane (with $x_3 = 0$). If $\mathbf{w}$ is a linear combination of $\mathbf{u}$ and $\mathbf{v}$, then $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is linearly dependent, by Theorem 7. Conversely, suppose that $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is linearly dependent. By Theorem 7, some vector in $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is a linear combination of the preceding vectors (since $\mathbf{u} \neq \mathbf{0}$). That vector must be $\mathbf{w}$, since $\mathbf{v}$ is not a multiple of $\mathbf{u}$. So $\mathbf{w}$ is in Span $\{\mathbf{u}, \mathbf{v}\}$. See Fig. 2.

Linearly dependent, w in Span{u, v}

Linearly independent, w not in Span{u, v}

FIGURE 2 Linear dependence in $\mathbb{R}^3$.

1.7 Linear Independence 59

Example 4 generalizes to any set $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ in $\mathbb{R}^3$ with $\mathbf{u}$ and $\mathbf{v}$ linearly independent. The set $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ will be linearly dependent if and only if $\mathbf{w}$ is in the plane spanned by $\mathbf{u}$ and $\mathbf{v}$.

The next two theorems describe special cases in which the linear dependence of a set is automatic. Moreover, Theorem 8 will be a key result for work in later chapters.

THEOREM 8

If a set contains more vectors than there are entries in each vector, then the set is linearly dependent. That is, any set $\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ in $\mathbb{R}^n$ is linearly dependent if $p > n$.

FIGURE 3 If $p > n$, the columns are linearly dependent.

PROOF Let $A = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_p]$. Then A is $n \times p$, and the equation $A\mathbf{x} = \mathbf{0}$ corresponds to a system of n equations in p unknowns. If $p > n$, there are more variables than equations, so there must be a free variable. Hence $A\mathbf{x} = \mathbf{0}$ has a nontrivial solution, and the columns of A are linearly dependent. See Fig. 3 for a matrix version of this theorem. ■

Warning: Theorem 8 says nothing about the case in which the number of vectors in the set does *not* exceed the number of entries in each vector.

FIGURE 4 A linearly dependent set in $\mathbb{R}^2$.

EXAMPLE 5 The vectors $\begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ -1 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix}$ are linearly dependent by Theorem 8, because there are three vectors in the set and there are only two entries in each vector. Notice, however, that none of the vectors is a multiple of one of the other vectors. See Fig. 4. ■

THEOREM 9

If a set $S = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ in $\mathbb{R}^n$ contains the zero vector, then the set is linearly dependent.

PROOF By renumbering the vectors, we may suppose $\mathbf{v}_1 = \mathbf{0}$. Then the equation $1\mathbf{v}_1 + 0\mathbf{v}_2 + \cdots + 0\mathbf{v}_p = \mathbf{0}$ shows that S is linearly dependent. ■

EXAMPLE 6 Determine by inspection if the given set is linearly dependent.

a. $\begin{bmatrix} 1 \\ 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 9 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix}, \begin{bmatrix} 4 \\ 1 \\ 8 \end{bmatrix}$ b. $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 8 \end{bmatrix}$ c. $\begin{bmatrix} -2 \\ 4 \\ 6 \\ 10 \end{bmatrix}, \begin{bmatrix} 3 \\ -6 \\ -9 \\ 15 \end{bmatrix}$

SOLUTION

a. The set contains four vectors, each of which has only three entries. So the set is linearly dependent by Theorem 8.

b. Theorem 8 does not apply here because the number of vectors does not exceed the number of entries in each vector. Since the zero vector is in the set, the set is linearly dependent by Theorem 9.

c. Compare the corresponding entries of the two vectors. The second vector seems to be $-3/2$ times the first vector. This relation holds for the first three pairs of entries, but fails for the fourth pair. Thus neither of the vectors is a multiple of the other, and hence they are linearly independent. ■

60 CHAPTER 1 Linear Equations in Linear Algebra

SG Mastering: Linear Independence 1–31

In general, you should read a section thoroughly *several* times to absorb an important concept such as linear independence. The notes in the *Study Guide* for this section will help you learn to form mental images of key ideas in linear algebra. For instance, the following proof is worth reading carefully because it shows how the definition of linear independence can be *used*.

PROOF OF THEOREM 7 (Characterization of Linearly Dependent Sets)

If some $\mathbf{v}_j$ in S equals a linear combination of the other vectors, then $\mathbf{v}_j$ can be subtracted from both sides of the equation, producing a linear dependence relation with a nonzero weight (-1) on $\mathbf{v}_j$. [For instance, if $\mathbf{v}_1 = c_2\mathbf{v}_2 + c_3\mathbf{v}_3$, then $\mathbf{0} = (-1)\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 + 0\mathbf{v}_4 + \cdots + 0\mathbf{v}_p$.] Thus S is linearly dependent.

Conversely, suppose S is linearly dependent. If $\mathbf{v}_1$ is zero, then it is a (trivial) linear combination of the other vectors in S. Otherwise, $\mathbf{v}_1 \neq \mathbf{0}$, and there exist weights $c_1, \dots, c_p$, not all zero, such that

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_p\mathbf{v}_p = \mathbf{0}$$

Let j be the largest subscript for which $c_j \neq 0$. If $j = 1$, then $c_1\mathbf{v}_1 = \mathbf{0}$, which is impossible because $\mathbf{v}_1 \neq \mathbf{0}$. So $j > 1$, and

$$c_1\mathbf{v}_1 + \cdots + c_j\mathbf{v}_j + 0\mathbf{v}_{j+1} + \cdots + 0\mathbf{v}_p = \mathbf{0}$$
$$c_j\mathbf{v}_j = -c_1\mathbf{v}_1 - \cdots - c_{j-1}\mathbf{v}_{j-1}$$
$$\mathbf{v}_j = \left(-\frac{c_1}{c_j}\right)\mathbf{v}_1 + \cdots + \left(-\frac{c_{j-1}}{c_j}\right)\mathbf{v}_{j-1}$$ ■

PRACTICE PROBLEMS

Let $\mathbf{u} = \begin{bmatrix} 3 \\ 2 \\ -4 \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} -6 \\ 1 \\ 7 \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} 0 \\ -5 \\ 2 \end{bmatrix}$, and $\mathbf{z} = \begin{bmatrix} 3 \\ 7 \\ -5 \end{bmatrix}$.

1. Are the sets $\{\mathbf{u}, \mathbf{v}\}$, $\{\mathbf{u}, \mathbf{w}\}$, $\{\mathbf{u}, \mathbf{z}\}$, $\{\mathbf{v}, \mathbf{w}\}$, $\{\mathbf{v}, \mathbf{z}\}$, and $\{\mathbf{w}, \mathbf{z}\}$ each linearly independent? Why or why not?
2. Does the answer to Problem 1 imply that $\{\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{z}\}$ is linearly independent?
3. To determine if $\{\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{z}\}$ is linearly dependent, is it wise to check if, say, $\mathbf{w}$ is a linear combination of $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{z}$?
4. Is $\{\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{z}\}$ linearly dependent?

1.7 EXERCISES

In Exercises 1–4, determine if the vectors are linearly independent. Justify each answer.

1. $\begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 7 \\ 2 \\ -6 \end{bmatrix}, \begin{bmatrix} 9 \\ 4 \\ -8 \end{bmatrix}$ 2. $\begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -8 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ 1 \end{bmatrix}$

3. $\begin{bmatrix} 2 \\ -3 \end{bmatrix}, \begin{bmatrix} -4 \\ 6 \end{bmatrix}$ 4. $\begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} -3 \\ -9 \end{bmatrix}$

In Exercises 5–8, determine if the columns of the matrix form a linearly independent set. Justify each answer.

5. $\begin{bmatrix} 0 & -3 & 9 \\ 2 & 1 & -7 \\ -1 & 4 & -5 \\ 1 & -4 & -2 \end{bmatrix}$ 6. $\begin{bmatrix} -4 & -3 & 0 \\ 0 & -1 & 5 \\ 1 & 1 & -5 \\ 2 & 1 & -10 \end{bmatrix}$

7. $\begin{bmatrix} 1 & 4 & -3 & 0 \\ -2 & -7 & 5 & 1 \\ -4 & -5 & 7 & 5 \end{bmatrix}$ 8. $\begin{bmatrix} 1 & -2 & 3 & 2 \\ -2 & 4 & -6 & 2 \\ 0 & 1 & -1 & 3 \end{bmatrix}$

In Exercises 9 and 10, (a) for what values of h is $\mathbf{v}_3$ in Span $\{\mathbf{v}_1, \mathbf{v}_2\}$, and (b) for what values of h is $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ linearly *dependent*? Justify each answer.

1.7 Linear Independence 61

9. $\mathbf{v}_1 = \begin{bmatrix} 1 \\ -3 \\ 2 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} -3 \\ 9 \\ -6 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} 5 \\ -7 \\ h \end{bmatrix}$

10. $\mathbf{v}_1 = \begin{bmatrix} 1 \\ -3 \\ -5 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} -3 \\ 9 \\ 15 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} 2 \\ -5 \\ h \end{bmatrix}$

In Exercises 11–14, find the value(s) of h for which the vectors are linearly *dependent*. Justify each answer.

11. $\begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ -6 \\ 7 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \\ h \end{bmatrix}$ 12. $\begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}, \begin{bmatrix} -6 \\ 4 \\ -3 \end{bmatrix}, \begin{bmatrix} 9 \\ h \\ 3 \end{bmatrix}$

13. $\begin{bmatrix} 1 \\ 5 \\ -3 \end{bmatrix}, \begin{bmatrix} -2 \\ -9 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ h \\ -9 \end{bmatrix}$ 14. $\begin{bmatrix} 1 \\ -2 \\ -4 \end{bmatrix}, \begin{bmatrix} -3 \\ 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ h \end{bmatrix}$

Determine by inspection whether the vectors in Exercises 15–20 are linearly *independent*. Justify each answer.

15. $\begin{bmatrix} 5 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 8 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ 7 \end{bmatrix}$ 16. $\begin{bmatrix} 2 \\ -4 \\ 8 \end{bmatrix}, \begin{bmatrix} -3 \\ 6 \\ -12 \end{bmatrix}$

17. $\begin{bmatrix} 5 \\ -3 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -7 \\ 2 \\ 4 \end{bmatrix}$ 18. $\begin{bmatrix} 3 \\ 4 \end{bmatrix}, \begin{bmatrix} -1 \\ 5 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 7 \\ 1 \end{bmatrix}$

19. $\begin{bmatrix} -8 \\ 12 \\ -4 \end{bmatrix}, \begin{bmatrix} 2 \\ -3 \\ -1 \end{bmatrix}$ 20. $\begin{bmatrix} 1 \\ 4 \\ -7 \end{bmatrix}, \begin{bmatrix} -2 \\ 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

In Exercises 21 and 22, mark each statement True or False. Justify each answer on the basis of a careful reading of the text.

21. a. The columns of a matrix A are linearly independent if the equation $A\mathbf{x} = \mathbf{0}$ has the trivial solution.
    b. If S is a linearly dependent set, then each vector is a linear combination of the other vectors in S.
    c. The columns of any 4×5 matrix are linearly dependent.
    d. If $\mathbf{x}$ and $\mathbf{y}$ are linearly independent, and if $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is linearly dependent, then $\mathbf{z}$ is in Span $\{\mathbf{x}, \mathbf{y}\}$.
22. a. If $\mathbf{u}$ and $\mathbf{v}$ are linearly independent, and if $\mathbf{w}$ is in Span $\{\mathbf{u}, \mathbf{v}\}$, then $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is linearly dependent.
    b. If three vectors in $\mathbb{R}^3$ lie in the same plane in $\mathbb{R}^3$, then they are linearly dependent.
    c. If a set contains fewer vectors than there are entries in the vectors, then the set is linearly independent.
    d. If a set in $\mathbb{R}^n$ is linearly dependent, then the set contains more than n vectors.

In Exercises 23–26, describe the possible echelon forms of the matrix. Use the notation of Example 1 in Section 1.2.

23. A is a 2×2 matrix with linearly dependent columns.
24. A is a 3×3 matrix with linearly dependent columns.
25. A is a 4×2 matrix, $A = [\mathbf{a}_1 \ \ \mathbf{a}_2]$, and $\mathbf{a}_2$ is not a multiple of $\mathbf{a}_1$.
26. A is a 4×3 matrix, $A = [\mathbf{a}_1 \ \ \mathbf{a}_2 \ \ \mathbf{a}_3]$, such that $\{\mathbf{a}_1, \mathbf{a}_2\}$ is linearly independent and $\mathbf{a}_3$ is not in Span $\{\mathbf{a}_1, \mathbf{a}_2\}$.
27. How many pivot columns must a 6×4 matrix have if its columns are linearly independent? Why?
28. How many pivot columns must a 4×6 matrix have if its columns span $\mathbb{R}^4$? Why?
29. Construct 3×2 matrices A and B such that $A\mathbf{x} = \mathbf{0}$ has a nontrivial solution, but $B\mathbf{x} = \mathbf{0}$ has *only* the trivial solution.
30. a. Fill in the blank in the following statement: "If A is an $m \times n$ matrix, then the columns of A are linearly independent if and only if A has _____ pivot columns."
    b. Explain why the statement in (a) is true.

Exercises 31 and 32 should be solved without *performing row operations*. [*Hint:* Write $A\mathbf{x} = \mathbf{0}$ as a vector equation.]

31. Given $A = \begin{bmatrix} 2 & 3 & 5 \\ -5 & 1 & -8 \\ -3 & -1 & -4 \\ 1 & 0 & 1 \end{bmatrix}$, observe that the third column is the sum of the first two columns. Find a nontrivial solution of $A\mathbf{x} = \mathbf{0}$.
32. Given $A = \begin{bmatrix} 4 & 3 & -5 \\ -2 & -2 & 4 \\ -2 & -3 & 7 \end{bmatrix}$, observe that the first column minus three times the second column equals the third column. Find a nontrivial solution of $A\mathbf{x} = \mathbf{0}$.

Each statement in Exercises 33–38 is either true (in all cases) or false (for at least one example). If false, construct a specific example to show that the statement is not always true. Such an example is called a *counterexample* to the statement. If a statement is true, give a *justification*. (One specific example cannot explain why a statement is always true. You will have to do more work here than in Exercises 21 and 22.)

33. If $\mathbf{v}_1, \dots, \mathbf{v}_r$ are in $\mathbb{R}^n$ and $\mathbf{v}_1 = 2\mathbf{v}_2 + \mathbf{v}_3$, then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ is linearly dependent.
34. If $\mathbf{v}_1$ and $\mathbf{v}_2$ are in $\mathbb{R}^4$ and $\mathbf{v}_2$ is not a scalar multiple of $\mathbf{v}_1$, then $\{\mathbf{v}_1, \mathbf{v}_2\}$ is linearly independent.
35. If $\mathbf{v}_1, \dots, \mathbf{v}_r$ are in $\mathbb{R}^3$ and $\mathbf{v}_3 = \mathbf{0}$, then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5\}$ is linearly dependent.
36. If $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are in $\mathbb{R}^3$ and $\mathbf{v}_3$ is *not* a linear combination of $\mathbf{v}_1, \mathbf{v}_2$, then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly independent.
37. If $\mathbf{v}_1, \dots, \mathbf{v}_4$ are in $\mathbb{R}^4$ and $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly dependent, then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ is also linearly dependent.
38. If $\{\mathbf{v}_1, \dots, \mathbf{v}_4\}$ is a linearly independent set of vectors in $\mathbb{R}^4$, then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is also linearly independent. [*Hint:* Think about $x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + x_3\mathbf{v}_3 + 0 \cdot \mathbf{v}_4 = \mathbf{0}$.]
39. Suppose A is an $m \times n$ matrix with the property that for all $\mathbf{b}$ in $\mathbb{R}^m$ the equation $A\mathbf{x} = \mathbf{b}$ has at most one solution. Use the

Figure 48.6: The book will have a lot of examples and their solutions.

88 CHAPTER 1 Linear Equations in Linear Algebra

SOLUTION TO PRACTICE PROBLEM

$$A = \begin{bmatrix} 36 & 51 & 13 \\ 52 & 34 & 74 \\ 0 & 7 & 1.1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 33 \\ 45 \\ 3 \end{bmatrix}$$

CHAPTER 1 SUPPLEMENTARY EXERCISES

1. Mark each statement True or False. Justify each answer. (If true, cite appropriate facts or theorems. If false, explain why or give a counterexample that shows why the statement is not true in every case.

a. Every matrix is row equivalent to a unique matrix in echelon form.

b. Any system of n linear equations in n variables has at most n solutions.

c. If a system of linear equations has two different solutions, it must have infinitely many solutions.

d. If a system of linear equations has no free variables, then it has a unique solution.

e. If an augmented matrix $[\,A \quad \mathbf{b}\,]$ is transformed into $[\,C \quad \mathbf{d}\,]$ by elementary row operations, then the equations $A\mathbf{x} = \mathbf{b}$ and $C\mathbf{x} = \mathbf{d}$ have exactly the same solution sets.

f. If a system $A\mathbf{x} = \mathbf{b}$ has more than one solution, then so does the system $A\mathbf{x} = \mathbf{0}$.

g. If A is an $m \times n$ matrix and the equation $A\mathbf{x} = \mathbf{b}$ is consistent for some $\mathbf{b}$, then the columns of A span $\mathbb{R}^m$.

h. If an augmented matrix $[\,A \quad \mathbf{b}\,]$ can be transformed by elementary row operations into reduced echelon form, then the equation $A\mathbf{x} = \mathbf{b}$ is consistent.

i. If matrices A and B are row equivalent, they have the same reduced echelon form.

j. The equation $A\mathbf{x} = \mathbf{0}$ has the trivial solution if and only if there are no free variables.

k. If A is an $m \times n$ matrix and the equation $A\mathbf{x} = \mathbf{b}$ is consistent for every $\mathbf{b}$ in $\mathbb{R}^m$, then A has m pivot columns.

l. If an $m \times n$ matrix A has a pivot position in every row, then the equation $A\mathbf{x} = \mathbf{b}$ has a unique solution for each $\mathbf{b}$ in $\mathbb{R}^m$.

m. If an $n \times n$ matrix A has n pivot positions, then the reduced echelon form of A is the $n \times n$ identity matrix.

n. If 3×3 matrices A and B each have three pivot positions, then A can be transformed into B by elementary row operations.

o. If A is an $m \times n$ matrix, if the equation $A\mathbf{x} = \mathbf{b}$ has at least two different solutions, and if the equation $A\mathbf{x} = \mathbf{c}$ is consistent, then the equation $A\mathbf{x} = \mathbf{c}$ has many solutions.

p. If A and B are row equivalent $m \times n$ matrices and if the columns of A span $\mathbb{R}^m$, then so do the columns of B.

q. If none of the vectors in the set $S = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in $\mathbb{R}^3$ is a multiple of one of the other vectors, then S is linearly independent.

r. If $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ is linearly independent, then $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are not in $\mathbb{R}^2$.

s. In some cases, it is possible for four vectors to span $\mathbb{R}^3$.

t. If $\mathbf{u}$ and $\mathbf{v}$ are in $\mathbb{R}^m$, then $-\mathbf{u}$ is in Span$\{\mathbf{u}, \mathbf{v}\}$.

u. If $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are nonzero vectors in $\mathbb{R}^2$, then $\mathbf{w}$ is a linear combination of $\mathbf{u}$ and $\mathbf{v}$.

v. If $\mathbf{w}$ is a linear combination of $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^3$, then $\mathbf{u}$ is a linear combination of $\mathbf{v}$ and $\mathbf{w}$.

w. Suppose that $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ are in $\mathbb{R}^3$, $\mathbf{v}_3$ is not a multiple of $\mathbf{v}_1$, and $\mathbf{v}_2$ is not a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_3$. Then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly independent.

x. A linear transformation is a function.

y. If A is a 6×5 matrix, the linear transformation $\mathbf{x} \mapsto A\mathbf{x}$ cannot map $\mathbb{R}^5$ onto $\mathbb{R}^6$.

z. If A is an $m \times n$ matrix with m pivot columns, then the linear transformation $\mathbf{x} \mapsto A\mathbf{x}$ is a one-to-one mapping.

2. Let a and b represent real numbers. Describe the possible solution sets of the (linear) equation $ax = b$. [*Hint:* The number of solutions depends upon a and b.]

3. The solutions (x, y, z) of a single linear equation

$$ax + by + cz = d$$

form a plane in $\mathbb{R}^3$ when a, b, and c are not all zero. Construct sets of three linear equations whose graphs (a) intersect in a single line, (b) intersect in a single point, and (c) have no points in common. Typical graphs are illustrated in the figure.

Three planes intersecting in a line (a)

Three planes intersecting at a point (b)

Three planes with no intersection (c)

Three planes with no intersection (c′)

4. Suppose the coefficient matrix of a linear system of three equations in three variables has a pivot position in each column. Explain why the system has a unique solution.

5. Determine h and k such that the solution set of the system (i) is empty, (ii) contains a unique solution, and (iii) contains infinitely many solutions.

a. $x_1 + 3x_2 = k$, $4x_1 + hx_2 = 8$

b. $-2x_1 + hx_2 = 1$, $6x_1 + kx_2 = -2$

6. Consider the problem of determining whether the following system of equations is consistent:

$$4x_1 - 2x_2 + 7x_3 = -5$$
$$8x_1 - 3x_2 + 10x_3 = -3$$

a. Define appropriate vectors, and restate the problem in terms of linear combinations. Then solve that problem.

b. Define an appropriate matrix, and restate the problem using the phrase "columns of A."

c. Define an appropriate linear transformation T using the matrix in (b), and restate the problem in terms of T.

7. Consider the problem of determining whether the following system of equations is consistent for all b_1, b_2, b_3:

$$2x_1 - 4x_2 - 2x_3 = b_1$$
$$-5x_1 + x_2 + x_3 = b_2$$
$$7x_1 - 5x_2 - 3x_3 = b_3$$

a. Define appropriate vectors, and restate the problem in terms of Span $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. Then solve that problem.

b. Define an appropriate matrix, and restate the problem using the phrase "columns of A."

c. Define an appropriate linear transformation T using the matrix in (b), and restate the problem in terms of T.

8. Describe the possible echelon forms of the matrix A. Use the notation of Example 1 in Section 1.2.

a. A is a 2×3 matrix whose columns span $\mathbb{R}^2$.

b. A is a 3×3 matrix whose columns span $\mathbb{R}^3$.

9. Write the vector $\begin{bmatrix} 5 \\ 6 \end{bmatrix}$ as the sum of two vectors, one on the line $\{(x, y) : y = 2x\}$ and one on the line $\{(x, y) : y = x/2\}$.

10. Let $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{b}$ be the vectors in $\mathbb{R}^2$ shown in the figure, and let $A = [\mathbf{a}_1 \quad \mathbf{a}_2]$. Does the equation $A\mathbf{x} = \mathbf{b}$ have a solution? If so, is the solution unique? Explain.

11. Construct a 2×3 matrix A, not in echelon form, such that the solution of $A\mathbf{x} = \mathbf{0}$ is a line in $\mathbb{R}^3$.

12. Construct a 2×3 matrix A, not in echelon form, such that the solution of $A\mathbf{x} = \mathbf{0}$ is a plane in $\mathbb{R}^3$.

13. Write the *reduced* echelon form of a 3×3 matrix A such that the first two columns of A are pivot columns and

$$A \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

14. Determine the value(s) of a such that $\left\{ \begin{bmatrix} 1 \\ a \end{bmatrix}, \begin{bmatrix} a \\ a+2 \end{bmatrix} \right\}$ is linearly independent.

15. In (a) and (b), suppose the vectors are linearly independent. What can you say about the numbers $a, \ldots, f$? Justify your answers. [*Hint:* Use a theorem for (b).]

a. $\begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b \\ c \\ 0 \end{bmatrix}, \begin{bmatrix} d \\ e \\ f \end{bmatrix}$

b. $\begin{bmatrix} a \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b \\ c \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} d \\ e \\ f \\ 1 \end{bmatrix}$

16. Use Theorem 7 in Section 1.7 to explain why the columns of the matrix A are linearly independent.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 5 & 0 & 0 \\ 3 & 6 & 8 & 0 \\ 4 & 7 & 9 & 10 \end{bmatrix}$$

17. Explain why a set $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ in $\mathbb{R}^3$ must be linearly independent when $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly independent and $\mathbf{v}_4$ is not in Span $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$.

18. Suppose $\{\mathbf{v}_1, \mathbf{v}_2\}$ is a linearly independent set in $\mathbb{R}^n$. Show that $\{\mathbf{v}_1, \mathbf{v}_1 + \mathbf{v}_2\}$ is also linearly independent.

Figure 48.7: The book will have a lot of examples and their solutions.

462 CHAPTER 8 The Geometry of Vector Spaces

If f is a nonzero functional, then rank $A = 1$, and dim Nul $A = n - 1$, by the Rank Theorem.[2] Thus, the subspace $[f : 0]$ has dimension $n - 1$ and so is a hyperplane. Also, if d is any number in $\mathbb{R}$, then

$$[f : d] \quad \text{is the same as} \quad \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = d\} \tag{2}$$

Recall from Theorem 6 in Section 1.5 that the set of solutions of $A\mathbf{x} = \mathbf{b}$ is obtained by translating the solution set of $A\mathbf{x} = \mathbf{0}$, using any particular solution $\mathbf{p}$ of $A\mathbf{x} = \mathbf{b}$. When A is the standard matrix of the transformation f, this theorem says that

$$[f : d] = [f : 0] + \mathbf{p} \quad \text{for any } \mathbf{p} \text{ in } [f : d] \tag{3}$$

Thus the sets $[f : d]$ are hyperplanes parallel to $[f : 0]$. See Fig. 1.

FIGURE 1 Parallel hyperplanes, with $f(\mathbf{p}) = d$.

When A is a $1 \times n$ matrix, the equation $A\mathbf{x} = d$ may be written with an inner product $\mathbf{n} \cdot \mathbf{x}$, using $\mathbf{n}$ in $\mathbb{R}^n$ with the same entries as A. Thus, from (2),

$$[f : d] \quad \text{is the same as} \quad \{\mathbf{x} \in \mathbb{R}^n : \mathbf{n} \cdot \mathbf{x} = d\} \tag{4}$$

Then $[f : 0] = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{n} \cdot \mathbf{x} = 0\}$, which shows that $[f : 0]$ is the orthogonal complement of the subspace spanned by $\mathbf{n}$. In the terminology of calculus and geometry for $\mathbb{R}^3$, $\mathbf{n}$ is called a **normal** vector to $[f : 0]$. (A "normal" vector in this sense need not have unit length.) Also, $\mathbf{n}$ is said to be **normal** to each parallel hyperplane $[f : d]$, even though $\mathbf{n} \cdot \mathbf{x}$ is not zero when $d \neq 0$.

Another name for $[f : d]$ is a *level set* of f, and $\mathbf{n}$ is sometimes called the *gradient* of f when $f(\mathbf{x}) = \mathbf{n} \cdot \mathbf{x}$ for each $\mathbf{x}$.

EXAMPLE 3 Let $\mathbf{n} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1 \\ -6 \end{bmatrix}$, and let $H = \{\mathbf{x} : \mathbf{n} \cdot \mathbf{x} = 12\}$, so $H = [f : 12]$, where $f(x, y) = 3x + 4y$. Thus H is the line $3x + 4y = 12$. Find an implicit description of the parallel hyperplane (line) $H_1 = H + \mathbf{v}$.

SOLUTION First, find a point $\mathbf{p}$ in H_1. To do this, find a point in H and add $\mathbf{v}$ to it. For instance, $\begin{bmatrix} 0 \\ 3 \end{bmatrix}$ is in H, so $\mathbf{p} = \begin{bmatrix} 1 \\ -6 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$ is in H_1. Now, compute $\mathbf{n} \cdot \mathbf{p} = -9$. This shows that $H_1 = [f : -9]$. See Fig. 2, which also shows the subspace $H_0 = \{\mathbf{x} : \mathbf{n} \cdot \mathbf{x} = 0\}$. ■

The next three examples show connections between implicit and explicit descriptions of hyperplanes. Example 4 begins with an implicit form.

[2] See Theorem 14 in Section 2.9 or Theorem 14 in Section 4.6.

8.4 Hyperplanes 463

FIGURE 2

EXAMPLE 4 In $\mathbb{R}^3$, give an explicit description of the line $x - 4y = 13$ in parametric vector form.

SOLUTION This amounts to solving a nonhomogeneous equation $A\mathbf{x} = \mathbf{b}$, where $A = [\,1 \quad -4\,]$ and $\mathbf{b}$ is the number 13 in $\mathbb{R}$. Write $x = 13 + 4y$, where y is a free variable. In parametric form, the solution is

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 13 + 4y \\ y \end{bmatrix} = \begin{bmatrix} 13 \\ 0 \end{bmatrix} + y\begin{bmatrix} 4 \\ 1 \end{bmatrix} = \mathbf{p} + y\mathbf{q}, \quad y \in \mathbb{R}$$ ■

Converting an explicit description of a line into implicit form is more involved. The basic idea is to construct $[f : 0]$ and then find d for $[f : d]$.

EXAMPLE 5 Let $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$, and let L_1 be the line through $\mathbf{v}_1$ and $\mathbf{v}_2$. Find a linear functional f and a constant d such that $L_1 = [f : d]$.

SOLUTION The line L_1 is parallel to the translated line L_0 through $\mathbf{v}_2 - \mathbf{v}_1$ and the origin. The defining equation for L_0 has the form

$$[\,a \quad b\,]\begin{bmatrix} x \\ y \end{bmatrix} = 0 \quad \text{or} \quad \mathbf{n} \cdot \mathbf{x} = 0, \quad \text{where} \quad \mathbf{n} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{5}$$

Since $\mathbf{n}$ is orthogonal to the subspace L_0, which contains $\mathbf{v}_2 - \mathbf{v}_1$, compute

$$\mathbf{v}_2 - \mathbf{v}_1 = \begin{bmatrix} 6 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \end{bmatrix}$$

and solve

$$[\,a \quad b\,]\begin{bmatrix} 5 \\ -2 \end{bmatrix} = 0$$

By inspection, a solution is $[\,a \quad b\,] = [\,2 \quad 5\,]$. Let $f(x, y) = 2x + 5y$. From (5), $L_0 = [f : 0]$, and $L_1 = [f : d]$ for some d. Since $\mathbf{v}_1$ is on line L_1, $d = f(\mathbf{v}_1) = 2(1) + 5(2) = 12$. Thus, the equation for L_1 is $2x + 5y = 12$. As a check, note that $f(\mathbf{v}_2) = f(6, 0) = 2(6) + 5(0) = 12$, so $\mathbf{v}_2$ is on L_1, too. ■

EXAMPLE 6 Let $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix}$, and $\mathbf{v}_3 = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$. Find an implicit description $[f : d]$ of the plane H_1 that passes through $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$.

Figure 48.8: The book will have a lot of examples and their solutions.

unumbered, so we will use normal LaTeX conventions to both define them as well as for author commands.

48.6 Geometry

Although we spend a good part of the Chapter on page design, reviewing historical typographical paper sizes, modern book production of text books is not bound with tradition but economics. High speed printing technology uses rolls and pages can be printed up to 64 pages at a time. We will follow the books dimensions which are 7.75x10.25in. The text area occupies approximately 0.67 of the textwidth and is particularly well balanced. Many mathematical text books come out too dense and are difficult to be used by students.

The Ecosystem

`phd` package and give a background as to how fonts are used in LaTeX.

h a n g

49

Key Value Interfaces

This chapter discusses primarily TEX's page breaking algorithm and how the output routines can be used to insert additional material on every page, such as inserts, page numbers and similar. Later the chapter discusses the terminology used in the TeXbook and by experts to describe the routines and we will develop some basic output routines. The last part touches on the main differences between LaTeX output routines and floats as compared to the original approach in PlainTeX.

21.00002
16
-2.5
-2.1
0.87865

The key value system greatly simplifies the TEX interface for authors. As **joseph2009** wrote this ease of use was not transferred into settting up key-value systems for authors of pre-packaged TEX code. This Chapter and the one that follows that focus specifically on the **pgfkeys** package provides an overview and describes some of the more difficult areas. The TUGboat article referenced earlier and written by Joseph Wright *et.al* has an excellent introduction to the available packages and some longer examples for comparison. Chapter **?? ??** discusses the `expl3` key-value functions.

49.1 keyval

The **keyval** written by David Carlisle is still widely used by package authors to provide the means for users to easily specify numerous optional arguments for macros **keyval**. The main advantages of using keyval are that (1) the number of optional arguments is no longer limited to 9 and that (2) the arguments are named, and hence there is less chance of confusion about the syntax of a macro.

49.2 xkeyval

A more recent package, **xkeyval** provides improvements for programming keys and also provides a more advanced interface for the namespacing of keys and families. Before you start experimenting with the xkeyval package, I suggest that you load the package **xkview**. This is part of the **xkeyval**[1] bundle and can help you to view key value parameters in various ways. The **xkeyval** package was developed by Hendri Adriaens and Uwe Kern[2]. This package is an extension of the well-known `keyval` package. The package provides more flexible commands and syntax enhancements as well as newer option processing mechanism.
The main change of the `xkeyval` package is that it provides a means to namespace the keys, which all have the form `\KV@family@keyname`, where the KV is a literal prefix to avoid collisions. They take one argument to handle user input.
The main commands of the package are the same as those of keyval.

Example 2.1 xkeyval

```
\makeatletter
\define@key{phd}{pi}{\setlength{\parindent}{#1}}
\setkeys{phd}{pi=50pt}
\makeatother
\lorem\par
\setkeys{phd}{pi=0pt}
\lorem\par
```

[1] http://www.ctan.org/pkg/xkeyval
[2] **xkeyval**.

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

Defining a default key, i.e., a key that can be used as `indent` or `indent=30pt` will stretch your memory, as it has an optional parameter as its third argument.

```
\define@key{family}{key}[none]{The input is: #1}
```

```
Example 2.2 xkeyval

\makeatletter

\define@key{phd}{pi}[30pt]{\setlength{\parindent}{#1}}

\setkeys{phd}{pi}

\lorem

or \setkeys{phd}{pi=0pt}

\lorem
\makeatother
```

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

or

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

49.3 Ordinary Keys

Ordinary keys are keys that have values such as `animal=elephant` and your macro can be called like `animals[animal=elephant]{14}`.

49.4 Keys and values in package options

First of all, the package supplies macros to declare class or package options, execute them and process them. The macros are available under the usual LaTeX names, but all with the suffix **X**, namely

`\DeclareOptionX`

`\DeclareOptionX*`

`\ExecuteOptionsX`

`\ProcessOptionsX`

These commands allow the user to assign a value to an option just like when using `\setkeys`. The first macro is based on `\define@key` and the final two are based on `\setkeys`. Supposing that a package `mypack` is set up with these commands, a user could for instance do

```
\usepackage[textcolor=red,font=times]{mypack}
```

These `xkeyval`macros are fully compatible with the LaTeX option conventions. They will allow packages to copy global options specified in the `\documentclass` command, to pass options to other classes or packages and to update the list of unused global options that will be displayed by LaTeX in the log file.

49.5 kvoptions

Another package **kvoptions** by Heiko Oberdiek is used extensively in the large suite of packages developed by Heiko[3]. The package originally formed part of the **hyperref** and later branched into an independent package. The package provides a number of additional commands to those found in the LaTeX kernel and a comparison of the commands is shown in the table below.
It is a good alternative that can be used for single topic package writing. An important feature of the package is its ability to process options both globally as well as locally avoiding conflicts when options are specified both globally as well as locally. Heiko provides an example from his bookmark package[4], which provides the option `open` that specifies whether the bookmarks are opened or closed initially. Its values are true or false. Since KOMA-Script version 3.00 the KOMA classes also introduces option open with values right and any and a complete different meaning. Such conflicts can be resolved by marking all or part of options as local by `\DeclareLocalOption` or `\DeclareLocalOptions`. Then the packages ignores global occurences of these options

[3] **kvoptions**.
[4] **bookmark**.

h a n g

50

Managing Keys with PGF

This chapter discusses primarily TEX's page breaking algorithm and how the output routines can be used to insert additional material on every page, such as inserts, page numbers and similar. Later the chapter discusses the terminology used in the TeXbook and by experts to describe the routines and we will develop some basic output routines. The last part touches on the main differences between LaTeX output routines and floats as compared to the original approach in PlainTeX.

21.00002
16
-2.5
-2.1
0.87865

50.1 PGF keys

This chapter describes the package **pgfkeys**. It is loaded automatically by both PGF and Ti*k*Z.
This package can be used independently of PGF. Note that no other package of PGF needs to be loaded (so neither the emulation layer nor the system layer is needed). The ConTEXt abbreviation is **pgfkey** if **pgfmod** is not loaded.

Example 1.1 wavelast

```
\bgroup
\cxset{paragraph style/uwave/font-weight=\itshape,
      % paragraph style/uwave/linewidth={15cm},
       }
\begin{center}

 \wavelast{
A Dialogue between the Landlady, and Susan the
    Chambermaid, proper to be read by all Innkeepers,
    and their Servants; with the Arrival, and affable
    Behaviour of a beautiful young Lady; which may
    teach Persons of Condition how they may acquire the
    Love of the whole World.}

\end{center}
\egroup
```

A Dialogue between the Landlady, and Susan the Chambermaid, proper to be read by all Innkeepers, and their Servants; with the Arrival, and affable Behaviour of a beautiful young Lady; which may teach Persons of Condition how they may acquire the Love of the whole World.

```
\def\wavelast#1{%
       \bgroup

       \setbox0=\vbox{\fontstore\noindent #1}%
       \setbox1=\vbox{%
            \unvbox0
            \setbox2=\lastbox
            \hbox to ⦇
                 ⦈10cm{\hfill\uwave{\unhbox2}\hfill}%
       }%
       \unvbox1
      \egroup
  }%
```

Consider the number of variables involved in such a paragraph. How can we abstract the code to make it more general? We can have settings for colors,

width, hyphenation, font related commands etc. PGF keys can be used to provide a dictionary.

A Dialogue between the Landlady, and Susan the Chambermaid, proper to be read by all Innkeepers, and their Servants; with the Arrival, and affable Behaviour of a beautiful young Lady; which may teach Persons of Condition how they may acquire the Love of the whole World.

Alice had never been in a court of justice before, but she had read about them in books, and she was quite pleased to find that she knew the name of nearly everything there. "That's the judge," she said to herself, "because of his great wig.". The judge, by the way, was the King, and as he wore his crown over the wig, (look at the frontispiece if you want to see how he did it,) he did not look at all comfortable, and it was certainly not becoming.

50.2 Introduction

50.2.1 Comparison to Other Packages

The `pgfkeys` package defines a key–value management system that is in some sense similar to the more light-weight `keyval` system and the improved `xkeyval` system. However, `pgfkeys` uses a slightly different philosophy than these systems and it will coexist peacefully with both of them. Its power and flexibility are evident in the `pgf` and `tikz` packages and to an extend our own `phd` system.
The main differences between `pgfkeys` and `xkeyval` are the following:

- `pgfkeys` organizes keys in a tree, while `keyval` and `xkeyval` use families. In `pgfkeys` the families correspond to the root entries of the key tree.
- For efficiency reasons, `pgfkeys` does not directly support setting keys drawn from multiple families as `xkeyval` does. This can be emulated if necessary, but it will be slower than `xkeyval`'s native support.
- `pgfkeys` has no save-stack impact (you will have to read the TEXBook very carefully to appreciate this).
- `pgfkeys` is slightly slower than `keyval`, but not much.
- `pgfkeys` supports styles. This means that keys can just stand for other keys (which can stand for other keys in turn or which can also just execute some code). Ti*k*Z uses this mechanism heavily.
- `pgfkeys` supports multi-argument key code. This can, however, be emulated in `keyval`.

- **pgfkeys** supports handlers. These are call-backs that are called when a key is not known. They are very flexible, in fact even defining keys in different ways is handled by, well, handlers.

The PGF can be used for complicated and multi-value cases. The fact that it is more capable than the rest does not in itself make it slower.

50.3 Technical

The package uses `\csname` to craete the names of macros and token registers to store the values. For example to store a length in a key `/test/length` it does the following:

Example 3.2 pgfkeyssetvalue

```
\pgfkeyssetvalue{/test/length}{2cm-3cm}
\makeatletter
\the\pgfkeys@temptoks

\csname pgfk@/test/length\endcsname
\makeatother
```

2cm-3cm
2cm-3cm

Essentially we defined a macro `pgfk@/test/length` with its replacement text being `2cm-3cm`. The definition of `\pgfkeyssetvalue` given by:

```
\long\def\pgfkeyssetvalue#1#2{%
  \pgfkeys@temptoks{#2}
  \expandafter
    \edef\csname ↲
          ↳pgfk@#1\endcsname{\the\pgfkeys@temptoks}%
}
```

The code above just stored a value. If the path ends in `.@cmd` it will set up things so the `key/.@cmd` contains a a macro that takes one parameter and has `#2` as its code.

```
\long\def\pgfkeysdef#1#2{%
  \long\def\pgfkeys@temp##1\pgfeov{#2}%
  \pgfkeyslet{#1/.@cmd}{\pgfkeys@temp}%
  \pgfkeyssetvalue{#1/.@body}{#2}%
}

\long\def\pgfkeysedef#1#2{%
  \long\edef\pgfkeys@temp##1\pgfeov{#2}%
  \pgfkeyslet{#1/.@cmd}{\pgfkeys@temp}%
  \pgfkeyssetvalue{#1/.@body}{#2}%
}
```

Example 3.3 Defining keys for code

```
\def\city#1{\expandafter\def\csname#1\endcsname{#1}}

\pgfkeysdef{/test/cities}{\city{#1}}

\pgfkeys{/test/cities=Rome}
\pgfkeys{/test/cities=Paris}

\Rome, \Paris
```

Rome, Paris

50.3.1 Getting started

The following quick guide to PGF's key mechanism only treats the most commonly used features. For an in-depth discussion of what is going on, please consult the PGF manual.

Keys are organized in a large tree that is similar to the Unix file tree. A typical key might be, say, `/tikz/coordinate system/x` or just `/x`. Again as in Unix, when you specify keys you can provide the complete path of the key, but you usually just provide the name of the key (corresponding to the file name without any path) and the path is added automatically.

Typically (but not necessarily) some code is associated with a key. To execute this code, you use the `\pgfkeys` command. This command takes a list of so-called key–value pairs. Each pair is of the form ⟨*key*⟩=⟨*value*⟩. For each pair the `\pgfkeys` command will execute the code stored for the ⟨*key*⟩ with its parameter set to ⟨*value*⟩.

Here is a typical example of how the `\pgfkeys` command is used:

```
\pgfkeys{/my key=hallo,/your keys/main key=something\strange,
         key name without path=something else}
```

Now, to set the code that is stored in a key you do not need to learn a new command. Rather, the `\pgfkeys` command can also be used to set the code of a key. This is done using so-called *handlers*. They look like keys whose names look like "hidden files in Unix" since they start with a dot. The handler for setting the code of a key is appropriately called `.code` and it is used as follows:

```
\pgfkeys{/my key/.code=The value is '#1'.}
\pgfkeys{/my key=hi!}
```

The value is 'hi!'.

As you can see, in the first line we defined the code for the key `/my key`. In the second line we executed this code with the parameter set to `hi!`.

There are numerous handlers for defining a key. For instance, we can also define a key whose value actually consists of more than one parameter.

```
\pgfkeys{/my key/.code 2 args=The values are '#1' and '#2'
\pgfkeys{/my key={a1}{a2}}
```

The values are 'a1' and 'a2'.

We often want to have keys where the code is called with some default value if the user does not provide a value. Not surprisingly, this is also done using a handler, this time called **.default**.

(hallo)(hello)
```
\pgfkeys{/my key/.code=(#1)}
\pgfkeys{/my key/.default=hello}
\pgfkeys{/my key=hallo,/my key}
```

The other way round, it is also possible to specify that a value *must* be specified, using a handler called **.value required**. Finally, you can also require that no value *may* be specified using **.value forbidden**.
All keys for a package like, say, Ti*k*Z start with the path **/tikz**. We obviously do not like to write this path down every time we use a key (so we do not have to write things like **\draw[/tikz/line width=1cm]**). What we need is to somehow "change the default path to a specific location." This is done using the handler **.cd** (for "change directory"). Once this handler has been used on a key, all subsequent keys *in the current call of* **\pgfkeys** *only* are automatically prefixed with this path, if necessary.
Here is an example:

```
\pgfkeys{/tikz/.cd,line width=1cm,line cap=round}
```

This makes it easy to define commands like **\tikzset**, which could be defined as follows (the actual definition is a bit faster, but the effect is the same):

```
\def\tikzset#1{\pgfkeys{/tikz/.cd,#1}}
```

When a key is handled, instead of executing some code, the key can also cause further keys to be executed. Such keys will be called *styles*. A style is, in essence, just a key list that should be executed whenever the style is executed. Here is an example:

(a:foo)(b:bar)(a:wow)
```
\pgfkeys{/a/.code=(a:#1)}
\pgfkeys{/b/.code=(b:#1)}
\pgfkeys{/my style/.style={/a=foo,/b=bar,/a=#1}}
\pgfkeys{/my style=wow}
```

As the above example shows, style can also be parametrized, just like the normal code keys.
Printing a phd key

Example 3.4 Printing a Key

```
\pgfkeys{/phd/test/.code=(Test),}
\pgfkeys{/phd/.style =/phd/.cd,
         phd, chapter name,
         }
\expandafter\meaning\expandafter\pgfkeysvalueof{/phd/chapter font-family}
\makeatletter
\csname pgfkey@store\endcsname {test}
\makeatother
```

macro:#1->\csname pgfk@#1\endcsname /phd/chapter font-family test

As a typical use of styles, suppose we wish to setup the key **/tikz** so that it will change the default path to **/tikz**. This can be achieved as follows:

```
\pgfkeys{/tikz/.style=/tikz/.cd}
\pgfkeys{tikz,line width=1cm,draw=red}
```

Note that when **\pgfkeys** is executed, the default path is set to **/**. This means that the first **tikz** will be completed to **/tikz**. Then **/tikz** is a style and, thus, replaced by **/tikz/.cd**, which changes the default path to **/tikz**. Thus, the **line width** is correctly prefixed with **/tikz**.

50.3.2 The Key Tree

The **pgfkeys** package organizes keys in a so-called *key tree*. This tree will be familiar to anyone who has used a Unix operating system: A key is addressed by a path, which consists of different parts separated by slashes. A typical key might be **/tikz/line width** or just **/tikz** or something more complicated like **/tikz/cs/x/.store in**.

Let us fix some further terminology: Given a key like **/a/b/c**, we call the part leading up the last slash (**/a/b**) the *path* of the key. We call everything after the last slash (**c**) the *name* of the key (in a file system this would be the file name).

We do not always wish to specify keys completely. Instead, we usually specify only part of a key (typically only the name) and the *default path* is then added to the key at the front. So, when the default path is **/tikz** and you refer to the (partial) key **line width**, the actual key that is used is **/tikz/line width**.

There is a simple rule for deciding whether a key is a partial key or a full key: If it starts with a slash, then it is a full key and it is not modified; if it does not start with a slash, then the default path is automatically prefixed.

Note that the default path is not the same as a search path. In particular, the default path is just a single path. When a partial key is given, only this single default path is prefixed; **pgfkeys** does not try to lookup the key in different parts of a search path. It is, however, possible to emulate search paths, but a much more complicated mechanism must be used.

When you set keys (to be explained in a moment), you can freely mix partial and full keys and you can change the default path. This makes it possible to temporarily use keys from another part of the key tree (this turns out to be a very useful feature).

Each key (may) store some *tokens* and there exist commands, described below, for setting, getting, and changing the tokens stored in a key. However, you will only very seldom use these commands directly. Rather, the standard way of using keys is the **\pgfkeys** command or some command that uses it internally like, say, **\tikzset**. So, you may wish to skip the following commands and continue with the next subsection.

\pgfkeyssetvalue{⟨*full key*⟩}{⟨*token text*⟩}

Stores the ⟨*token text*⟩ in the ⟨*full key*⟩. The ⟨*full key*⟩ may not be a partial key, so no default-path-adding is done. The ⟨*token text*⟩ can be arbitrary tokens and may even contain things like # or unbalanced TEX-ifs.

Hello, world!

```
\pgfkeyssetvalue{/my family/my key}{Hello, world!}
\pgfkeysvalueof{/my family/my key}
```

The setting of a key is always local to the current TEX group.

Lorem ipsum ...
```
\pgfkeyssetvalue{/phd/chapteris font}{\arial}
{\pgfkeysvalueof{/phd/chapteris font}  Lorem ipsum ...}
```

\pgfkeyslet{⟨*full key*⟩}{⟨*macro*⟩}

Performs a `\let` statement so the the ⟨*full key*⟩ pionts to the contents of ⟨*macro*⟩.

Hello, world!
```
\def\helloworld{Hello, world!}
\pgfkeyslet{/my family/my key}{\helloworld}
\pgfkeysvalueof{/my family/my key}
```

You should never let a key be equal to `\relax`. Such a key may or may not be indistinguishable from an undefined key.

\pgfkeysgetvalue{⟨*full key*⟩}{⟨*macro*⟩}

Retrieves the tokens stored in the ⟨*full key*⟩ and lets ⟨*macro*⟩ be equal to these tokens. If the key has not been set, the ⟨*macro*⟩ will be equal to `\relax`.

Hello, world!
```
\pgfkeyssetvalue{/my family/my key}{Hello, world!}
\pgfkeysgetvalue{/my family/my key}{\helloworld}
\helloworld
```

\pgfkeysvalueof{⟨*full key*⟩}

Inserts the value stored in ⟨*full key*⟩ at the current position into the text.

Hello, world!
```
\pgfkeyssetvalue{/my family/my key}{Hello, world!}
\pgfkeysvalueof{/my family/my key}
```

\pgfkeysifdefined{⟨*full key*⟩}{⟨*if*⟩}{⟨*else*⟩}

Checks whether this key was previously set using either `\pgfkeyssetvalue` or `\pgfkeyslet`. If so, the code in ⟨*if*⟩ is executed, otherwise the code in ⟨*else*⟩.
This command will use eTEX's `\ifcsname` command, if available, for efficiency. This means, however, that it may behave differently for TEX and for eTEX when you set keys to `\relax`. For this reason you should not do so.

yes
```
\pgfkeyssetvalue{/my family/my key}{Hello, world!}
\pgfkeysifdefined{/my family/my key}{yes}{no}
```

50.3.3 Setting Keys

Settings keys is done using a powerful command called `\pgfkeys`. This command takes a list of so-called *key–value pairs*. These are pairs of the form ⟨*key*⟩=⟨*value*⟩. The principle idea is the following: For each pair in the list, some *action* is taken. This action can be one of the following:

1. A command is executed whose argument(s) are ⟨*value*⟩. This command is stored in a special subkey of ⟨*key*⟩.
2. The ⟨*value*⟩ is stored in the ⟨*key*⟩ itself.
3. If the key's name (the part after the last slahs) is a known *handler*, then this handler will take care of the key.
4. If the key is totally unknown, one of several possible *unknown key handlers* is called.

Addtionally, if the ⟨*value*⟩ is missing, a default value may or may not be substituted. Before we plunge into all the details, let us have a quick look at the command itself.

`\pgfkeys`{⟨*key list*⟩}

The ⟨*key list*⟩ should be a list of key–value pairs, separated by commas. A key–value pair can have the following two forms: ⟨*key*⟩=⟨*value*⟩ or just ⟨*key*⟩. Any spaces around the ⟨*key*⟩ or around the ⟨*value*⟩ are removed. It is permissible to surround both the ⟨*key*⟩ or the ⟨*value*⟩ in curly braces, which are also removed. Especially putting the ⟨*value*⟩ in curly braces needs to be done quite often, namely whenever the ⟨*value*⟩ contains an equal-sign or a comma.

The key–value pairs in the list are handled in the order they appear. How this handling is done, exactly, is described in the rest of this section.

If a ⟨*key*⟩ is a partial key, the current value of the default path is prefixed to the ⟨*key*⟩ and this "upgraded" key is then used. The default path is just the root path `/` when the first key is handled, but it may change later on. At the end of the command, the default path is reset to the value it had before this command was executed.

Calls of this command may be nested. Thus, it is permissible to call `\pgfkeys` inside the code that is executed for a key. Since the default path is restored after a call of `\pgfkeys`, the default path will not change when you call `\pgfkeys` while executing code for a key (which is exactly what you want).

`\pgfqkeys`{⟨*default path*⟩}{⟨*key list*⟩}

This command has the same effect as `\pgfkeys`{⟨*default path*⟩`/.cd,`⟨*key list*⟩}, it is only marginally quicker. This command should not be used in user code, but rather in commands like `\tikzset` or `\pgfset` that get called very often.

`\pgfkeysalso`{⟨*key list*⟩}

This command has execatly the same effect as `\pgfkeys`, only the default path is not modified before or after the keys are being set. This command is mainly intended to be called by the code that is being processed for a key.

`\pgfqkeysalso`{⟨*default path*⟩}{⟨*key list*⟩}

This command has the same effect as `\pgfkeysalso`{⟨*default path*⟩`/.cd,`⟨*key list*⟩}, it is only quicker. Changing the default path inside a `\pgfkeyalso` is dangerous, so use with care. A rather safe place to call this command is at the beginning of a TEX group.

50.3.30 Default Arguments

The arguments of the `\pgfkeys` command can either be of the form ⟨*key*⟩=⟨*value*⟩ or of the form ⟨*key*⟩ with the value-part missing. In the second case, the `\pgfkeys` will try to provide a *default value* for the ⟨*value*⟩. If such a default value is defined, it will be used as if you had written ⟨*key*⟩=⟨*default value*⟩.
In the following, the details of how default values are determined is described; however, you should normally use the handlers `.default` and `.value required` as described in Section **??** and you can may wish to skip the following details.
When `\pgfkeys` encounters a ⟨*key*⟩ without an equal-sign, the following happens:

1. The input is replaced by ⟨*key*⟩=`\pgfkeysnovalue`. In particular, the commands `\pgfkeys{my key}` and `\pgfkeys{my key=\pgfkeysnovalue}` have exactly the same effect and you can "simulate" a missing value by providing the value `\pgfkeysnovalue`, which is sometimes useful.

2. If the ⟨*value*⟩ is `\pgfkeysnovalue`, then it is checked whether the subkey ⟨*key*⟩`/.@def` exists. For instance, if you write `\pgfkeys{/my key}`, then it is checked whether the key `/my key/.@def` exists.

3. If the key ⟨*key*⟩`/.@def` exists, then the tokens stored in this key are used as ⟨*value*⟩.

4. If the key does not exist, then `\pgfkeysnovalue` is used as the ⟨*value*⟩.

5. At the end, if the ⟨*value*⟩ is now equal to `\pgfkeysvaluerequired`, then the code (or something fairly equivalent) `\pgfkeys{/errors/value required=`⟨*key*⟩`{}}` is executed. Thus, by changing this key you can change the error message that is printed or you can handle the missing value in some other way.

50.3.4 Keys That Execute Commands

After the transformation process described in the previous subsection, we arrive at a key of the form ⟨*key*⟩=⟨*value*⟩, where ⟨*key*⟩ is a full key. Different things can now happen, but always the macro `\pgfkeyscurrentkey` will have been setup to expand to the text of the ⟨*key*⟩ that is currently being processed.
The first things that is tested is whether the key ⟨*key*⟩`/.@cmd` exists. If this is the case, then it is assumed that this key stores the code of a macro and this macro is executed. The argument of this macro is ⟨*value*⟩ directly followed by `\pgfeov`, which stands for "end of value." The ⟨*value*⟩ is not surrounded by braces. After this code has been executed, `\pgfkeys` continues with the next key in the ⟨*key list*⟩.
It may seem quite peculiar that the macro stored in the key ⟨*key*⟩`/.@cmd` is not simply executed with the argument `{`⟨*value*⟩`}`. However, the approach taken in the `pgfkeys` packages allows for more flexibility. For instance, assume that

you have a key that expects a ⟨*value*⟩ of the form "⟨*text*⟩+⟨*more text*⟩" and wishes to store ⟨*text*⟩ and ⟨*more text*⟩ in two different macros. This can be achieved as follows:

Example 3.5 Storing values in macros

```
{ }

\def\mystore#1+#2\pgfeov{\def\a{#1}\def\b{#2}}
\pgfkeyslet{/my key/.@cmd}{\mystore}
\pgfkeys{/my key=hello+world}

|\a| is \a, |\b| is \b.
```

`\a` is hello, `\b` is world.

Naturally, defining the code to be stored in a key in the above manner is too awkward. A simpler approach is shown in Example **??**. The following commands simplify things a bit, but the usual manner of setting up code for a key is to use one of the handlers described in Section **??**.

`\pgfkeysdef`{⟨*key*⟩} {⟨*code*⟩}

This command temporarily defines a TEX-macro with the argument list `#1\pgfeov` and then lets ⟨*key*⟩`/.@cmd` be equal to this macro. The net effect of all this is that you have then setup code for the key ⟨*key*⟩ so that when you write `\pgfkeys{`⟨*key*⟩`=`⟨*value*⟩`}`, then the ⟨*code*⟩ is executed with all occurrences of `#1` in ⟨*code*⟩ being replaced by ⟨*value*⟩. (This behaviour is quite similar to the `\define@key` command of `keyval` and `xkeyval`).

`\pgfkeysedef`{⟨*key*⟩} {⟨*code*⟩}

This command works like `\pgfkeysdef`, but it uses `\edef` rather than `\def` when defining the key macro. If you do not know the difference between the two, then you will not need this command; and if you know the difference, then you will know when you need this command.

`\pgfkeysdeargs`{⟨*key*⟩}{⟨*argument pattern*⟩}{⟨*code*⟩}

This command works like `\pgfkeysdef`, but it allows you to provide an arbitrary ⟨*argument pattern*⟩ rather than just the simple single argument of `\pgfkeysdef`.

Example 3.6 Delineated arguments

```
{}
\pgfkeysdefargs{/my key}{#1+#2}{\def\a{#1}\def\b{#2}}
\pgfkeys{/my key=hello+world}

|\a| is \a, |\b| is \b.
```

> \a is hello, \b is world.

\pgfkeysedefargs{⟨*key*⟩}{⟨*argument pattern*⟩}{⟨*code*⟩}

The \edef version of \pgfkeysdefargs.

50.3.5 Keys That Store Values

Let us continue with what happens when \pgfkeys processes the current key and the subkey ⟨*key*⟩/.@cmd is not defined. Then it is checked whether the ⟨*key*⟩ itself exists (has been previously assigned a value using, for instance, \pgfkeyssetvalue). In this case, the tokens stored in ⟨*key*⟩ are replaced by ⟨*value*⟩ and \pgfkeys proceeds with the next key in the ⟨*key list*⟩.

50.3.6 Keys That Are Handled

If neither the ⟨*key*⟩ itself nor the subkey ⟨*key*⟩/.@cmd are defined, then the ⟨*key*⟩ cannot be processed "all by itself." Rather, a ⟨*handler*⟩ is needed for this key. Most of the power of pgfkeys comes from the proper use of such handlers. Recall that the ⟨*key*⟩ is always a full key (if it was not originally, it has already been upgraded at this point to a full key). It decomposed into two parts:

1. The ⟨*path*⟩ of ⟨*key*⟩ (everything before the last slash) is stored in the macro \pgfkeyscurrentpath.

2. The ⟨*name*⟩ of ⟨*key*⟩ (everything after the last slash) is stored in the macro \pgfkeyscurrentname.

   It is recommended (but not necessary) that the name of a handler starts with a dot (but not with .@), so that they are easy to detect for the reader.

(For efficiency reasons, these two macros are only setup at this point; so when code is executed for a key in the "usual" manner then these macros are not setup.)
The \pgfkeys command now checks whether the key /handlers/⟨*name*⟩/.@cmd exists. If so, it should store a command and this command is executed exactly in the same manner as described in Section ??. Thus, this code gets the ⟨*value*⟩ that was originally intended for ⟨*key*⟩ as its argument, followed by \pgfeov. It is the job of the handlers to so something useful with the ⟨*value*⟩.
For an example, let us write a handler that will output the value stored in a key to the log file. We call this handler .print to log. The idea is that when someone tries to use the key /my key/.print to log, then this key will not be defined and the handler gets executed. The handler will then have access to the path-part of the key, which is /my key, via the macro \pgfkeyscurrentpath. It can then lookup which value is stored in this key and print it.

```
\pgfkeysdef{/handlers/.print to log}
{%
  \pgfkeysgetvalue{\pgfkeyscurrentpath}{\temp}
  \writetolog{\temp}
}
\pgfkeyssetvalue{/my key}{Hi!}
...
\pgfkeys{/my key/.print to log}

\pgfkeys{/phd/chapter label/.font-size}

font sise
```

The above code will print `Hi!` in the log, provided the macro `\writetolog` is setup appropriately.
For a more interesting handler, let us program a handler that will setup a key so that when the key is used some code is executed. This code is given as ⟨*value*⟩. All the handler must do is to call `\pgfkeysdef` for the path of the key (which misses the handler's name) and assign the parameter value to it.

Example 3.7 Defining Handlers

```
\pgfkeysdef{/handlers/.my code}
      {\pgfkeysdef{\pgfkeyscurrentpath}{#1}}
\pgfkeys{/my key/.my code=(#1)}
\pgfkeys{/my key=hallo}
```

(hallo)

50.3.7 Keys That Are Unknown

For some keys, neither the key is defined nor its `.@cmd` subkey nor is a handler defined for this key. In this case, it is checked whether the key ⟨*current path*⟩`/.unknown/.@cmd` exists. Thus, when you try to use the key `/tikz/strange`, then it is checked whether `/tikz/.unknown/.@cmd` exists. If this key exists (which it does), it is executed. This code can then try to make sense of the key. For instance, the handler for Ti*k*Z will try to interpret the key's name as a color or as an arrow specification or as a PGF option.
You can setup unknown key handlers for your own keys by simply setting the code of the key ⟨*my path prefix*⟩`/.unknown`. This also allows you to setup "search paths." The idea is that you would like keys to be searched not only in a single default path, but in several. Suppose, for instance, that you would like keys to be searched for in `/a`, `/b`, and `/b/c`. We setup a key `/my search path` for this:

```
\pgfkeys{/my search path/.unknown/.code=
  {%
    \let\searchname=\pgfkeyscurrentname%
    \pgfkeysalso{%
```

```
      /a/\searchname/.try=#1,
      /b/\searchname/.retry=#1,
      /b/c/\searchname/.retry=#1%
    }%
  }%
}
\pgfkeys{/my search path/.cd,foo,bar}
```

In the above code, **foo** and **bar** will be searched for in the three directories **/a**, **/b**, and **/b/c**.
If the key ⟨*current path*⟩**/.unknown/.@cmd** does not exist, the handler **/handlers/.unknown** is invoked instead, which is always defined and which prints an error message by default.

50.3.8 Key Handlers

We now describe which key handlers are defined by default. You can also define new ones as described in Section **??**.

50.3.9 Handlers for Path Management

Key handler ⟨*key*⟩**/.cd**

This handler causes the default path to be set to ⟨*key*⟩. Note that the default path is reset at the beginning of each call to **\pgfkeys** to be equal to **/**.

Key handler ⟨*key*⟩**/.is family**

This handler sets up things such that when ⟨*key*⟩ is executed, then the current path is set to ⟨*key*⟩. A typical use is the following:

```
\pgfkeys{/tikz/.is family}
\pgfkeys{tikz,line width=1cm}
```

The effect of this handler is the same as if you had written ⟨*key*⟩**/.style=**⟨*key*⟩**/.cd**, only the code produced by the **.is family** handler is quicker.
This is an importnat handler if you are going to write your own library or package. It will enable you to write code of the form **\mypackageset{}** allowing your users not to have to type every time the full path.

Example 3.8 The .is family handler

```
\def\pkgfamilyname{phd}
\pgfkeys{/\pkgfamilyname/.is family}
%
%\newcommand\cxset{\pgfqkeys{/\pkgfamilyname}}

\def\cxkeydef#1#2{%
 \pgfkeyssetvalue{/\pkgfamilyname/#1}{#2}%
}
```

```
\def\cxvalueof#1{%
 \pgfkeysvalueof{/\pkgfamilyname/#1}%
}

\pgfkeyssetvalue{/phd/test}{Hello World.}
\pgfkeysvalueof{/phd/test}\\

\cxset{/phd/test=Hello Another world.}
\pgfkeysvalueof{/phd/test}

\cxvalueof{test}

\let\phdvalueof\cxvalueof
\let\phdset\cxset

\phdset{test=Hello PHD Worlds}
\phdvalueof{test}
```

Hello World.
Hello Another world.
Hello Another world.
Hello PHD Worlds

50.3.10 Setting Defaults

Key handler ⟨*key*⟩/`.default`

=⟨*value*⟩ Sets the default value of ⟨*key*⟩ to ⟨*value*⟩. This means that whenever no value is provided in a call to `\pgfkeys`, then this ⟨*value*⟩ will be used instead. This still means taht we need to have the key defined earlier The key needs to be initialized first.

```
Example 3.9 Using default

\pgfkeys{myfamily/color/.initial =red, }
\pgfkeys{myfamily/color}
```

red

Key handler ⟨*key*⟩/`.value required`

This handler causes the error message key `/errors/value required` to be issued whenever the ⟨*key*⟩ is used without a value.

Example: `\pgfkeys{/width/.value required}`

Key handler ⟨*key*⟩/`.value forbidden`

This handler causes the error message key `/erros/value forbidden` to be issued whenever the ⟨*key*⟩ is used with a value.

This handler works be adding code to the code of the key. This means that you have to define the key first before you can use this handler.

```
\pgfkeys{/my key/.code=I do not want an argument!}
\pgfkeys{/my key/.value forbidden}

\pgfkeys{/my key}     % Ok
\pgfkeys{/my key=foo} % Error
```

50.3.11 Defining Key Codes

A number of handlers exist for defining the code of keys.

Key handler ⟨*key*⟩/`.code`

=⟨*code*⟩
This handler executes `\pgfkeysdef` with the parameters ⟨*key*⟩ and ⟨*code*⟩. This means that, afterwards, whenever the ⟨*key*⟩ is used, the ⟨*code*⟩ gets executed. More precisely, when ⟨*key*⟩=⟨*value*⟩ is encountered in a key list, ⟨*code*⟩ is executed with any occurrence of `#1` replaced by ⟨*value*⟩. As always, if no ⟨*value*⟩ is given, the default value is used, if defined, or the special value `\pgfkeysnovalue`.
It is permissible that ⟨*code*⟩ calls the command `\pgfkeys`. It is also permissible the ⟨*code*⟩ calls the command `\pgfkeysalso`, which is useful for styles, see below.

```
  Example 3.10 Some test

\pgfkeys{/par indent/.code={\parindent=#1},/par
    indent/.default=2em}
\pgfkeys{/par indent=1cm}
...
\pgfkeys{/par indent}

    ...
```

Key handler ⟨*key*⟩/`.ecode`

=⟨*code*⟩ This handler works like `.code`, only the command `\pgfkeysedef` is used.

Key handler ⟨*key*⟩/`.code 2 args`

=⟨*code*⟩ This handler works like `.code`, only two arguments rather than one are expected when the ⟨*code*⟩ is executed. This means that when ⟨*key*⟩=⟨*value*⟩ is encountered in a key list, the ⟨*value*⟩ should consist of two arguments. For instance, ⟨*value*⟩ could be `{first}{second}`. Then ⟨*code*⟩ is executed with any occurrence of `#1` replaced `first` and any occurrence of `#2` replaced by `second`.

Because of the special way the ⟨*value*⟩ is parsed, if you set ⟨*value*⟩ to, for instance, `first` (without any braces), then `#1` will be set to `f` and `#2` will be set to `irst`.

Key handler ⟨*key*⟩/`.ecode 2 args`

=⟨*code*⟩
This handler works like `.code 2 args`, only an `\edef` is used rather than a `\def` to define the macro.

Key handler ⟨*key*⟩/`.code args`

={⟨*argument pattern*⟩}{⟨*code*⟩} This handler also works like `.code`, but you can now specify an arbitrary ⟨*argument pattern*⟩. Such a pattern is a usual TEX macro pattern. For instance, suppose ⟨*argument pattern*⟩ is `(#1/#2)` and ⟨*key*⟩=⟨*value*⟩ is encountered in a key list with ⟨*value*⟩ being `(first/second)`. Then ⟨*code*⟩ is executed with any occurrence of `#1` replaced `first` and any occurrence of `#2` replaced by `second`. So, the actual ⟨*value*⟩ is matched against the ⟨*argument pattern*⟩ in the standard TEX way.

Key handler ⟨*key*⟩/`.ecode args={`⟨***argument pattern***⟩`}{`⟨***code***⟩`}`

This handler works like `.code args`, only an `\edef` is used rather than a `\def` to define the macro.
There are also handlers for modifying existing keys.

Key handler ⟨*key*⟩/`.add code={`⟨***prefix code***⟩`}{`⟨***append code***⟩`}`

This handler adds code to an existing key. The ⟨*prefix code*⟩ is added to the code stored in ⟨*key*⟩`/.@cmd` at the beginning, the ⟨*append code*⟩ is added to this code at the end. Either can be empty. The argument list of ⟨*code*⟩ cannot be changed using this handler. Note that both ⟨*prefix code*⟩ and ⟨*append code*⟩ may contain parameters like `#2`.

```
\pgfkeys{/par indent/.code={\parindent=#1}}
\newdimen\myparindent
\pgfkeys{/par indent/.add code={}{\myparindent=#1}}
...
\pgfkeys{/par indent=1cm} % This will set both \parindent and
                          % \myparindent to 1cm
```

Key handler ⟨*key*⟩/`.prefix code=`⟨***prefix code***⟩

This handler is a shortcut for ⟨*key*⟩`/.add code={`⟨*prefix code*⟩`}{}`. That is, this handler adds the ⟨*prefix code*⟩ at the beginning of the code stored in ⟨*key*⟩`/.@cmd`.

Key handler ⟨*key*⟩/`.append code=`⟨***append code***⟩

This handler is a shortcut for ⟨*key*⟩`/.add code={}{`⟨*append code*⟩`}{}`.

50.3.12 Defining Styles

The following handlers allow you to define *styles*. A style is a key list that is processed whenever the style is given as a key in a key list. Thus, a style "stands for" a certain key value list. Styles can be parameterized just like normal code.

Key handler ⟨*key*⟩/`.style=`⟨***key list***⟩

This handler set things up so that whenever ⟨*key*⟩`=`⟨*value*⟩ is encountered in a key list, then the ⟨*key list*⟩, with every occurrence of `#1` replaced by ⟨*value*⟩, is processed instead. As always, if no ⟨*value*⟩ is given, the default value is used, if defined, or the special value `\pgfkeysnovalue`.
You can achieve the same effect by writing ⟨*key*⟩`/.code=\pgfkeysalso{`⟨*key list*⟩`}`. This means, in particular, that the code of a key could also first execute some normal code and only then process some further keys.

Key handler ⟨*key*⟩/`.estyle`

`=`⟨*key list*⟩
This handler works like `.style`, only the ⟨*code*⟩ is set using `\edef` rather than `\def`. Thus, all macros in the ⟨*code*⟩ are expanded prior to saving the style.
For styles the corresponding handlers as for normal code exist:

Key handler ⟨*key*⟩/`.style 2 args=`

⟨*key list*⟩ This handler works like `.code 2 args`, only for styles. Thus, the ⟨*key list*⟩ may contain occurrences of both `#1` and `#2` and when the style is used, two parameters must be given as ⟨*value*⟩.

```
Example 3.11 example

\pgfkeys{/paper height/.code={\paperheight=#1},/paper
    width/.code={\paperwidth=#1}}
\pgfkeys{/page size/.style 2 args={/paper
    height=#1,/paper width=#2}}
\pgfkeys{/page size={30cm}{20cm}}
```

Key handler ⟨*key*⟩/`.estyle 2 args`

`=`⟨*key list*⟩ This handler works like `.style 2 args`, only an `\edef` is used rather than a `\def` to define the macro.

Key handler ⟨*key*⟩/`.style args`

`={`⟨*argument pattern*⟩`}{`⟨*key list*⟩`}` This handler works like `.code args`, only for styles.

Key handler ⟨*key*⟩/`.estyle args`

={⟨*argument pattern*⟩}{⟨*code*⟩} This handler works like **.ecode args**, only for styles.

Key handler ⟨*key*⟩/**.add style**

={⟨*prefix key list*⟩}{⟨*append key list*⟩}
This handler works like **.add code**, only for styles. However, it is permissible to add styles to keys that have previously been set using **.code**. (It is also permissible to add normal ⟨*code*⟩ to a key that has previously been set using **.style**). When you add a style to a key that was previously set using **.code**, the following happens: When ⟨*key*⟩ is processed, the ⟨*prefix key list*⟩ will be processed first, then the ⟨*code*⟩ that was previously stored in ⟨*key*⟩**/.@cmd**, and then the keys in ⟨*append key list*⟩ are processed.

```
\pgfkeys{/par indent/.code={\parindent=#1}}
\pgfkeys{/par indent/.add style={}{/my key=#1}}
...
\pgfkeys{/par indent=1cm} % This will set \parindent and
                          % then execute /my key=#1
```

Key handler ⟨*key*⟩/**.prefix style**

=⟨*prefix key list*⟩ Works like **.add style**, but only for the prefix key list.

Key handler ⟨*key*⟩/**.append style**

=⟨*append key list*⟩ Works like **.add style**, but only for the append key list.

50.3.120 Defining Value-, Macro-, If- and Choice-Keys

For some keys, the code that should be executed for them is rather "specialized." For instance, it happens often that the code for a key just sets a certain TEX-if to true or false. For these case predefine handlers make it easier to install the necessary code.
However, we start with some handlers that are used to manage the value that is directly stored in a key.

Key handler ⟨*key*⟩/**.initial**

=⟨*value*⟩ This handler sets the value of ⟨*key*⟩ to ⟨*value*⟩. Note that no subkeys are involved. After this handler has been used, by the rules governing keys, you can subsequently change the value of the ⟨*key*⟩ by just writing ⟨*key*⟩=⟨*value*⟩. Thus, this handler is used to set the initial value of key.
Note that in the after the example, writing **\pgfkeys{/my key}** will not have the effect you might expect (namely that **blue** is inserted into the main text). Rather, **/my key** will be promoted to **/my key=\pgfkeysnovalue** and, thus, **\pgfkeysnovalue** will be stored in **/my key**.
To retrieve the value stored in a key, the handler **.get** is used.

Example 3.12 The .initial handler

```
\pgfkeys{/my key/.initial=red}
% "/my key" now stores the value "red"
\pgfkeys{/my key=blue}
% "/my key" now stores the value "blue"
\pgfkeys{/my key}
```

blue

Key handler ⟨*key*⟩/`.get`

=⟨*macro*⟩ Executes a `\let` command so that ⟨*macro*⟩ contains the contents stored in ⟨*key*⟩.

Example 3.13 The handler .get

```
\pgfkeys{/my key/.initial=red}
\pgfkeys{/my key=blue}
\pgfkeys{/my key/.get=\mymacro}
\mymacro
```

blue

Key handler ⟨*key*⟩/`.add`

={⟨*prefix value*⟩}{⟨*append value*⟩} Adds the ⟨*prefix value*⟩ and the beginning and the ⟨*append value*⟩ at the end of the value stored in ⟨*key*⟩.
The next handler is useful for the common situation where ⟨*key*⟩=⟨*value*⟩ should cause the ⟨*value*⟩ to be stored in some macro. Note that, typically, you could just as well store the value in the key itself.

Key handler ⟨*key*⟩/`.store in`

=⟨*macro*⟩ This handler has the following effect: When you write ⟨*key*⟩=⟨*value*⟩, the code `\def`⟨*macro*⟩{⟨*value*⟩} is executed. Thus, the given value is "stored" in the ⟨*macro*⟩.

Hello Gruffalo!

```
\pgfkeys{/text/.store in=\mytext}
\def\a{world}
\pgfkeys{/text=Hello \a!}
\def\a{Gruffalo}
\mytext
```

Key handler ⟨*key*⟩/`.estore in`

=⟨*macro*⟩ This handler is similar to `.store in`, only the code `\edef`⟨*macro*⟩{⟨*value*⟩} is used. Thus, the macro-expanded version of ⟨*value*⟩ is stored in the ⟨*macro*⟩.

Hello world!

```
\pgfkeys{/text/.estore in=\mytext}
\def\a{world}
\pgfkeys{/text=Hello \a!}
\def\a{Gruffalo}
\mytext
```

The definition of this handler is trivial:

```
\pgfkeys{/handlers/.store in/.code=
  \pgfkeysalso{\pgfkeyscurrentpath/.code=\def#1{##1}}}
```

How about a csname one?

Example 3.14 Defining a new handler

```
\pgfkeys{/handlers/.csname in/.code=%
   \pgfkeysalso{\pgfkeyscurrentpath/.code=
     \expandafter\def\csname#1\endcsname{##1}}}
\pgfkeys{/my test/fonts/.csname in=font-store}
\pgfkeys{/my test/fonts=\bfseries}
\bgroup
\csuse{font-store} This is bold text.
\egroup
```

This is bold text.

If we wanted the user never to have to type a command, but only strings we could do something like this:

Example 3.15 Defining a new handler

```
\pgfkeys{/handlers/.csstore in/.code=%
   \pgfkeysalso{\pgfkeyscurrentpath/.code=
     \expandafter\def\csname#1\endcsname{\csname##1\endcsname}}}
\pgfkeys{/my test/fonts/.csstore in=font-store}
\pgfkeys{/my test/fonts=bfseries}
\bgroup
\csuse{font-store} This is bold text.
\egroup
```

This is bold text.

We can take an idea from another great library **expl3** this time from the LaTeX 3 Team. We will modify the above code and create a handler to choose a font-weight from a list of predefined keywords. For example we may want to use css-like font-weights.

Example 3.16 using Expl3 to Define handlers

```
\ExplSyntaxOn
\pgfkeys{/handlers/.fontweights/.code =
  \pgfkeysalso
    {\pgfkeyscurrentpath/.code=
      \tl_set:Nn\l_tmpa_str:N {##1}
         \str_case_x:nnTF {##1}
           {
             { none     } { \cs_gset:cpn {#1} {\mdseries } }
             { bold     } { \cs_gset:cpn {#1} {\bfseries } }
             { normal   } { \cs_gset:cpn {#1} {\mdseries } }
             { bfseries } { \cs_gset:cpn {#1} {\bfseries } }
             { mdseries } { \cs_gset:cpn {#1} {\mdseries } }
           }
           {                                }
           { \cs_gset:cpn {#1} {\mdseries} }
    }
 }
\ExplSyntaxOff
% 1  define where the values will be stored
\pgfkeys{/my test/font weight/.fontweights = font-weight-store}

% 2 store a value
\pgfkeys{/my test/font weight = bold}

% 3 use the store to set the weight
\csuse{font-weight-store} This is bold weight text.

% type an unknown value
\pgfkeys{/my test/font weight=very strong}

% this will just give medium series

\csuse{font-weight-store} This is medium weight text.
```

This is bold weight text.
This is medium weight text.

We can extend the code to give us an error message, if we use a value that is not acceptable. We could also have written the code using the `.choice` handler. I personally think that using custom defined handlers is a much better option as it can make part of our code portable to other packages (rather than cut-and-paste).

Of course LaTeX comes with its own set of key generation macros. Personally I made the choice to go this route, for most of the **PHD** package. I made the choice

in order to offer a consistent user interface that consists only of one command to set the keys, that does not require any knowlwdge of LaTeX commands to set document parameters.
In another common situation a key is used to set a TeX-if to true or false.

Key handler ⟨*key*⟩/`.is if`

=⟨*TeX-if name*⟩ This handler has the following effect: When you write ⟨*key*⟩=⟨*value*⟩, it is first checked that ⟨*value*⟩ is `true` or `false` (the default is `true` if no ⟨*value*⟩ is given). If this is not the case, the error key `/errors/boolean expected` is executed. Otherwise, the code `\`⟨*TeX-if name*⟩⟨*value*⟩ is executed, which sets the TeX-if accordingly.

Round?

```
\newif\iftheworldisflat
\pgfkeys{/flat world/.is if=theworldisflat}
\pgfkeys{/flat world=false}
\iftheworldisflat
  Flat
\else
  Round?
\fi
```

The next handler deals with the problem when a ⟨*key*⟩=⟨*value*⟩ makes sense only for a small set of possible ⟨*value*⟩s. For instance, the line cap can only be `rounded` or `rect` or `butt`, but nothing else. For this situation the following handler is useful.

Key handler ⟨*key*⟩/`.is choice`

This handler set things up so that writing ⟨*key*⟩=⟨*value*⟩ will cause the subkey ⟨*key*⟩/⟨*value*⟩ to be executed. So, each of the different possible choices should be given by a subkey of ⟨*key*⟩.

```
\pgfkeys{/line cap/.is choice}
\pgfkeys{/line cap/round/.style={\pgfsetbuttcap}}
\pgfkeys{/line cap/butt/.style={\pgfsetroundcap}}
\pgfkeys{/line cap/rect/.style={\pgfsetrectcap}}
\pgfkeys{/line cap/rectangle/.style={/line cap=rect}}
...
\draw [/line cap=butt] ...
```

If the subkey ⟨*key*⟩/⟨*value*⟩ does not exist, the error key `/errors/unknown choice value` is executed.
Behind the scenes PGF uses some mind-boggling code:

```
\pgfkeys{/handlers/.is choice/.code=%
  \pgfkeys{%
    \pgfkeyscurrentpath/.cd,%
    .code=\def\pgfkeys@was@choice{##1}%
       \expandafter\pgfkeysalso\expandafter{\pgfkeyscurrentkey/##1},
    .unknown/.code={%
      \def\pgf@marshal{\pgfkeysvalueof{/errors/unknown choice value/.@cmd}}%
      {\expandafter\expandafter\expandafter\pgf@marshal
          \expandafter\expandafter\expandafter{\expandafter
             \the\expandafter\pgfkeys@pathtoks\expandafter}
               \expandafter{\pgfkeys@was@choice}\pgfeov}%
    }%
  }%
}
```

50.3.120 Expanding Values

When you write ⟨*key*⟩=⟨*value*⟩, you usually wish to use the ⟨*value*⟩ "as is." Indeed, great care is taken to ensure that you can even use things like `#1` or unbalanced TEX-ifs inside ⟨*value*⟩. However, sometimes you want the ⟨*value*⟩ to be expanded before it is used. For instance, ⟨*value*⟩ might be a macro name like `\mymacro` and you do not want `\mymacro` to be used as the macro, but rather the *contents* of `\mymacro`. Thus, instead of using ⟨*value*⟩ you wish to use whatever ⟨*value*⟩ expands to. Instead of using some fancy `\expandafter` hackery, you can use the following handlers:

Key handler ⟨*key*⟩/`.expand once`

=⟨*value*⟩ This handler expands ⟨*value*⟩ once (more precisely, it executes an `\expandafter` command on the first token of ⟨*value*⟩) and then process the resulting ⟨*result*⟩ as if you had written ⟨*key*⟩=⟨*result*⟩. Note that if ⟨*key*⟩ contains a handler itself, this handler will be called normally.

```
Example 3.17 Handling expansion

\def\a{bottom}
\def\b{\a}
\def\c{\b}

\pgfkeys{/phd/key1/.initial=\c}
\pgfkeys{/phd/key2/.initial/.expand once=\c}
\pgfkeys{/phd/key3/.initial/.expand twice=\c}
\pgfkeys{/phd/key4/.initial/.expanded=\c}

\def\a{{\ttfamily\string\a}}
\def\b{{\ttfamily\string\b}}
\def\c{{\ttfamily\string\c}}

\begin{tabular}{ll}
Key 1:& \pgfkeys{/phd/key1} \\
```

```
Key 2:& \pgfkeys{/phd/key2} \\
Key 3:& \pgfkeys{/phd/key3} \\
Key 4:& \pgfkeys{/phd/key4}
\end{tabular}
```

Key 1:	\c
Key 2:	\b
Key 3:	\a
Key 4:	bottom

In the example above, you can observe how the key macro definitions can be expanded. When `.expanded` is used the macro is fully expanded (see key4), whereas in the other two expansion is controlled. These are mind-boggling and some patience and skill is required to use them effectively.

Key handler ⟨*key*⟩/`.expand twice`

=⟨*value*⟩ This handler works like saying ⟨*key*⟩`/.expand once/.expand once=`⟨*value*⟩.

Key handler ⟨*key*⟩/`.expanded`

=⟨*value*⟩ This handler will completely expand ⟨*value*⟩ (using `\edef`) before processing ⟨*key*⟩=⟨*result*⟩.

50.3.120 Handlers for Testing Keys

Key handler ⟨*key*⟩/`.try`

=⟨*value*⟩ This handler causes the same things to be done as if ⟨*key*⟩=⟨*value*⟩ had been written instead. However, if neither ⟨*key*⟩`/.@cmd` nor the key itself is defined, no handlers will be called. Instead, the execution of the key just stops. Thus, this handler will "try" to use the key, but no further action is taken when the key is not defined.
The TEX-if `\ifpgfkeyssuccess` will be set according to whether the ⟨*key*⟩ was successfully executed or not.

(a:hallo)(b:welt) text

```
\pgfkeys{/a/.code=(a:#1)}
\pgfkeys{/b/.code=(b:#1)}
\pgfkeys{/x/.try=blue,/a/.try=hallo,/b/.try=welt,/tikz/.tr
```

Key handler ⟨*key*⟩/`.retry`

=⟨*value*⟩ This handler works just like `.try`, only it will not do anything if `\ifpgfkeyssuccess` is false. Thus, this handler will only retry to set a key if "the last attempt failed".

(a:hallo)

```
\pgfkeys{/a/.code=(a:#1)}
\pgfkeys{/b/.code=(b:#1)}
\pgfkeys{/x/.try=hmm,/a/.retry=hallo,/b/.retry=welt}
```

50.3.120 Handlers for Key Inspection

Key handler ⟨*key*⟩/`.show value`

This handler executes a `\show` command on the value stored in ⟨*key*⟩. This is useful mostly for debugging.

Example: `\pgfkeys{/my/obscure key/.show value}`

Key handler ⟨*key*⟩/`.show code`

This handler executes a `\show` command on the code stored in ⟨*key*⟩`/.@cmd`. This is useful mostly for debugging.

Example: `\pgfkeys{/my/obscure key/.show code}`

The following key is not a handler, but it also commonly used for inspecting things:

`/utils/exec`=⟨*code*⟩

This key will simply execute the given ⟨*code*⟩.

Example: `\pgfkeys{some key=some value,/utils/exec=\show\hallo,obscure key=obscur`

50.3.13 Error Keys

In certain situations errors can occur, like using an undefined key. In these situations error keys are executed. They should store a macro that gets two arguments: The first is the offending key (possibly only after macro expansion), the second is the value that was passed as a parameter (also possibly only after macro expansion).

Currently, error keys are simply executed. In the future it might be a good idea to have different subkeys that are executed depending on the language currently set so that users get a localized error message.

`/errors/value required`={⟨*offending key*⟩}{⟨*value*⟩}

This key is executed whenever an ⟨*offending key*⟩ is used without a value when a value is actually required.

`/errors/value forbidden`={⟨*offending key*⟩}{⟨*value*⟩}

This key is executed whenever a key is used with a value when a value is actually forbidden.

`/errors/boolean expected`={⟨*offending key*⟩}{⟨*value*⟩}

This key is executed whenever a key setup using `.is if` gets called with a ⟨*value*⟩ other than `true` or `false`.

`/errors/unknown choice value`={⟨*offending key*⟩}{⟨*value*⟩}

This key is executed whenever a choice is used as a ⟨*value*⟩ for a key setup using the `.is choice` handler that is not defined.

`/errors/unknown key`={⟨*offending key*⟩}{⟨*value*⟩}`no default`

This key is executed whenever a key is unknown and no specific `.unknown` handler is found.

expl3

`phd` package and give a background as to how fonts are used in LaTeX.

A

phddoc User Manual

This chapter discusses primarily TEX's page breaking algorithm and how the output routines can be used to insert additional material on every page, such as inserts, page numbers and similar. Later the chapter discusses the terminology used in the TeXbook and by experts to describe the routines and we will develop some basic output routines. The last part touches on the main differences between LaTeX output routines and floats as compared to the original approach in PlainTeX.

21.00002

16

-2.5

-2.1

0.87865

A.1 Documentation of the LaTeX sources

This is a class for documenting the **phd** bundle, a collection of packages and classes that enables the typesetting of documents using a flexible user interface, You may however find it generally useful as a class for typesetting the documentation of files produced in 'doc' format.
The class is written as a "self-contained" docstrip file: executing `latex phddoc.dtx` generates the `phddoc.cls` file and typesets this documentation; execute `tex l3doc.dtx` to only generate `phddoc.cls`.
Each documented file in the standard distribution comes with extension `dtx`. The appropriate class package or initex file will be extracted from the source by the docstrip system. Each `dtx` file may be directly processed with LaTeX 2_ε, for example

```
% latex2e docclass.dtx
%
```

would produce the documentation of the Class and package interface.
Each file that is used in producing the LaTeX 2_ε format (ie not including the standard class and packages) will be printed together in one document if you LaTeX the file `sources2e.tex`. This has the advantage that one can produce a full index of macro usage across all the source files.
If you need to customise the typesetting of any of these files, there are two options:

- You can use DOCSTRIP with the module 'driver' to extract a small LaTeX file that you may edit to use whatever class or package options you require, before inputting the source file.
- You can create a file `phddoc.cfg`. This configuration file will be read whenever the `phddoc` class is used, and so can be used to customise the typesetting of all the source files, without having to edit lots of small driver files.

The second option is usually more convenient. Various possibilities are discussed in the next section.

A.2 Specification

The class builds on the **ltxdoc**[1] class and the **doc**[2] package, but since they were written many authors have come up with different ideas, as to how these documents should be produced.
The LaTeX3 Team has also more recently developed the `l3doc` class and `l3docstrip` package for documenting the l3 sources. Other Teams such as the developers of **pgf** prefer not to use `docstrip` and document the code and user manuals in a more traditional way, as normal documents in conjuction with external scripts written in python.
My objectives in writing this package, was to integrate the ability of the other packages in this series to document code in a flexible way. For longer books,

[1] Carlisle, David (Mar. 2018). *The file ltxdoc.dtx for use with LaTeX2e*.
[2] Mittelbach (2018).

such as a thesis, where the author might use their own developed macros, it also enables one to use such a method.
This class can be considered as a framework, as it can be used to produce almost any type of document.
The objectives are as follows:

Flexibilty Provide flexibility to use one of the standard LaTeX 2e classes `article`, `book report` or the `KOMA` classes `scrartcl`, `scrbook`, `scrreprt` as the main class. In addition to classes normally used for documents, the class also can be used for documenting LaTeX or LaTeX 3 packages and classes.

Style Enable the use of a fully featured key value interface for documenting the code.

Tools Provide a series of tools to create new documents, formatting and scaffolding. Currently LaTeX distributions have a plethora of tools, mostly using perl and lately l3build using Lua. Perl tools have served the community well for many years. One such tool `ctanify` does not work using normal dstributions as the Perl bundled in the distributions has some missiong modules. Go is a cross-compliation systems language enabling scripts to be bundled for different operating systems easily, hence the choice here.[3] Some of these problems with Perl on Windows can be overcome using `Strawberry Perl`[4] For any conflicts follow the guidelines in penwatch.[5]

```
phd ctanify  myclass.dtx  myclass.ins  README
```

A.3 Customisation

The simplest form of customisation is to pass more options to the `article` class which is loaded by **phddoc**. For instance if you wish all the documentation to be formated for A4 paper, add the following line to `phddoc.cfg`:

```
% \PassOptionsToClass{a4paper}{article}
%
```

All the source files are in two parts, separated by `\StopEventually`. The first part (should) contain 'user' documentation. The second part is a full documented listing of the source code. The `doc` package provides the command `\OnlyDescription` which suppresses the code listings. This may also be used in the configuration file, but as the `doc` package is read later, you must delay the execution of `\OnlyDescription` until after the `doc` package has been read. The simplest way is to use `\AtBeginDocument`. Thus you could put the following in your `phddoc.cfg`.

```
% \AtBeginDocument{\OnlyDescription}
```

[3] See for example https://tex.stackexchange.com/questions/256096/which-perl-to-install-for-xindy-with-miktex-on-windows

[4] Donload at http://strawberryperl.com/. This will also enable xindy to work on a MikTeX distribution.

[5] https://www.penwatch.net/cms/pip_conflict/

If the full source listing **sources2e.tex** is processed, then an index and change history are produced by default, however indices are not normally produced for individual files.
As an example, consider **ltclass.dtx**, which contains the sources for the new class and package interface commands. With no **cfg** file, a 19 page document is produced. With the above configuration a slightly more readable document (4 pages) is produced.
Conversely, if you really want to read the source listings in detail, you will want to have an index. Again the index commands provided by the doc package may be used, but their execution must be delayed.

```
% \AtBeginDocument{\CodelineIndex\EnableCrossrefs}
% \AtEndDocument{\PrintIndex}
```

The **doc** package writes index files to be sorted using MakeIndex with the **gind** style, so one would then use a command such as

```
% makeindex -s gind.ist ltclass.idx
%
```

and re-run LaTeX.
Similarly to print a Change history, you would add

```
% \AtBeginDocument{\RecordChanges}
% \AtEndDocument{\PrintChanges}
%
```

to **phddoc.cfg**, and use MakeIndex with a comand such as

```
% makeindex -s gglo.ist -o ltclass.gls ltclass.glo
%
```

Finally if you do not want to list all the sections of **source2e.tex**, you can use **\includeonly** in the **cfg** file:

```
% \includeonly{ltvers,ltboxes}
%
```

B

Implementation Code

This chapter discusses primarily TEX's page breaking algorithm and how the output routines can be used to insert additional material on every page, such as inserts, page numbers and similar. Later the chapter discusses the terminology used in the TeXbook and by experts to describe the routines and we will develop some basic output routines. The last part touches on the main differences between LaTeX output routines and floats as compared to the original approach in PlainTeX.

21.00002
16
-2.5
-2.1
0.87865

B.1 Options

```
*class

%\RequirePackage{underscore}

\ExplSyntaxOn
\cs_gset:Npn \l_phd_version{1.00}
\ExplSyntaxOff
```

1.00

rocode_colorize_bool Boolean to switch off the colorizing of code appearing within a macrocode environment.

Define the prefix of the module

```
@@=phdcl

\ExplSyntaxOn
\bool_new:N \g__phdcl_macrocode_colorize_bool
\bool_new:N \g__phdcl_book_bool
\bool_new:N \g__phdcl_article_bool
\bool_new:N \g__phdcl_report_bool
\bool_new:N \g__phdcl_scrbook_bool
\bool_new:N \g__phdcl_scrartcl_bool
\bool_new:N \g__phdcl_scrreprt_bool
\bool_new:N \g__phdcl_masterthesis_bool
\bool_new:N \g__phdcl_ldoc_bool
\bool_new:N \g__phdcl_ldociii_bool
\ExplSyntaxOff

\ExplSyntaxOn
\bool_if:NTF \g__phdcl_macrocode_colorize_bool
  {\bool_gset_true:N \phdd_code_colorize_bool}
  {\bool_gset_false:N \phdd_code_colorize_bool}
```

(*End definition for* `\g_@@_macrocode_colorize_bool`.)

B.1.1 Geometry parameters

```
\DeclareOption{a5paper}{\@latexerr{Option not supported}%
   {}}

\DeclareOption { full }
  {
    \bool_gset_true:N \g_phdd_typeset_documentation_bool
    \bool_gset_true:N \g_phdd_typeset_implementation_bool
  }
\DeclareOption { onlydoc }
  {
    \bool_gset_true:N \g_phdd_typeset_documentation_bool
    \bool_gset_false:N \g_phdd_typeset_implementation_bool
  }
\DeclareOption{colorize}
  {\bool_gset_true:N \phdd_code_colorize_bool}
```

```
\DeclareOption { check }
  { \bool_gset_true:N \g__phdcl_checkfunc_bool }
\DeclareOption { nocheck }
  { \bool_gset_false:N \g__phdcl_checkfunc_bool }
\DeclareOption { checktest }
  { \bool_gset_true:N \g__phdcl_checktest_bool }
\DeclareOption { nochecktest }
  { \bool_gset_false:N \g__phdcl_checktest_bool }
\DeclareOption { kernel }
  { \bool_gset_true:N \g__phdcl_kernel_bool }
\DeclareOption { stdmodule }
  { \bool_gset_false:N \g__phdcl_kernel_bool }
\DeclareOption { cm-default }
  { \bool_gset_false:N \g__phdcl_lmodern_bool }
\DeclareOption { lm-default }
  { \bool_gset_true:N \g__phdcl_lmodern_bool }
\DeclareOption { cs-break-off }
  { \bool_gset_false:N \g__phdcl_cs_break_bool }
\DeclareOption { cs-break-nohyphen }
  { \PassOptionsToPackage{nohyphen}{underscore} }
\tl_new:N \g__phdcl_doctype_tl
\DeclareOption {book}
  {\tl_gput_right:Nn \g__phdcl_doctype_tl{book} }
\DeclareOption {scrbook}
  {\tl_gput_right:Nn \g__phdcl_doctype_tl{book} }
\DeclareOption* { \PassOptionsToClass { \CurrentOption } { book } }
\ExecuteOptions { full, kernel, nocheck, nochecktest, lm-
default }
\PassOptionsToClass { a4paper } { book }
```

B.2 Configuration

Input a local configuration file, if it exists, with a message to the console that this has happened. Since we distribute a `.cfg` file with the class, this should usually always be true. Therefore, check for `\ExplMakeTitle` (defined in "our" `.cfg` file) and only output the informational message if it's not found.

```
\ExplSyntaxOn
\msg_new:nnn { phdcl } { input-cfg }
  { Local~config~file~phdcl.cfg~loaded. }
\file_if_exist:nTF {phdcl.cfg}
  {
    \file_input:n {phdcl.cfg}
  }
  { \msg_info:nn { phdcl } { input-cfg }  }
  {}
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\str_case_e:nnTF { \g__phdcl_doctype_tl }
   {
     { book     }    {
                        %\LoadClass{book}
                        \bool_gset_true:N \g__phdcl_book_bool

                      }
     { article  }    {
                        %\LoadClass{article}
                        \bool_gset_true:N \g__phdcl_article_bool
                      }
     { report        } {
      %\LoadClass{report}
      \bool_gset_true:N \g__phdcl_report_bool
       }
     { scrbook       } {
                          %\LoadClass{scrbook}
                          \KOMAoptions{twoside = false}
                          \bool_gset_true:N \g__phdcl_scrbook_bool
                        }
     { scrartcl      } {
                          %\LoadClass{scrartcl}
                          \bool_gset_true:N \g__phdcl_scrartcl_bool
                        }
     { l3doc         } { %\LoadClass{l3doc}
                        }
     { masterthesis  } {                          }
     { tufte         } {                          }
     { ltxdoc        } {                          }
     { l3doc         } {                          }
     { memoir        } { %\LoadClass{memoir}
                                                 }
   }
   {                                              }
   { }

\ExplSyntaxOff
```

B.3 Option Processing

```
\ExplSyntaxOn
\ProcessOptions
%\ProcessKeysPackageOptions
\LoadClass{\g__phdcl_doctype_tl}
\ExplSyntaxOff
```

B.4 Loading book and doc

The original **ltxdoc** uses the article class. For longer documentation it is preferable to use the book, so for the **phddoc** class I have opted to default it to book.

```
% hypdoc is loaded with the phd-packagemanager so that
% the right order for packages and patches can be provided
\RequirePackage{doc}
\RequirePackage{phd}
\RequirePackage{phd-pkgmanager}
\sethyperref
\RequirePackage{phd-documentation} %modifies doc as necessary
\RequirePackage{phd-colorpalette}
\RequirePackage{phd-runningheads}
\RequirePackage{phd-toc}

```

Make **|** be a 'short verb' character, but not in the document preamble, where an active character may interfere with packages that are loaded.

```
\AtBeginDocument{\MakeShortVerb{\|}}
```

As 'doc' documents tend to have a lot of monospaced material, Set up some **tt** substitutions to occur silently. [1] [2]

```
%\DeclareFontShape{OT1}{cmtt}{bx}{n}{<-> ssub * cmtt/m/n}{}
%\DeclareFontFamily{OMS}{cmtt}{\skewchar\font 48}  % '60
%\DeclareFontShape{OMS}{cmtt}{m}{n}{<-> ssub * cmsy/m/n}{}
%\DeclareFontShape{OMS}{cmtt}{bx}{n}{<-> ssub * cmsy/b/n}{}
```

This substitution is in the standard fd file, but not silent.

```
\DeclareFontShape{OT1}{cmss}{m}{it}{<->ssub*cmss/m/sl}{}
```

```
\CodelineIndex
\CodelineNumbered
\EnableCrossrefs
```

Increase the text width slightly so that width the standard fonts 72 columns of code may appear in a **macrocode** environment. [3]

```
%\setlength{\textwidth}{375pt}
```

Increase the marginpar width slightly, for long command names. And increase the left margin by a similar amount [4] [5]

```
%\addtolength\marginparwidth{40pt}
%\addtolength\oddsidemargin{40pt}
%\addtolength\evensidemargin{40pt}
```

```
\setcounter{StandardModuleDepth}{1}
```

[1] v2.0p,1995/11/02,Add font substitutions.
[2] v2.0t,1999/04/17,Replaced octal number, CAR.
[3] v2.0c,1994/03/15,Set **\textwidth**..
[4] v2.0l,1994/05/25,Increase **\marginparwidth**.
[5] v2.0q,1995/11/28,Increase **\marginparwidth** and page margin..

B.5 Useful abbreviations

The **phd-documentation** provides numerous commands for typesetting LaTeX code. It is imported automatically by the **phddoc** class and hence the following macros are described here for convenience.
`\cmd{\foo}` Prints `\foo` verbatim. It may be used inside moving arguments. It can *not* be use to record commands that are defined as "`\outer`" nor is it possible to use it on conditionals such as `\iftrue` or defined by `\newif`. `\cs{foo}` also prints `\foo`, for those who prefer that syntax. (This second form can be used to record all type of commends so the above restrictions do not apply.

`\cmd`
`\cs`

```
\def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}
\def\cmd@to@cs#1#2{\char\number`#2\relax}

\DeclareRobustCommand\cls{\textcolor{thered}}
%\newcommand\cs[1]{\color{blue}{\texttt{\char`\ #1}}}

```

phddoc

(*End definition for* `\cmd` *and* `\cs`.)

`\marg` `\marg{text}` prints {⟨*text*⟩}, 'mandatory argument'.

```
%\providecommand\marg[1]{%
%  {\ttfamily\char`\{}\meta{#1}{\ttfamily\char`\}}}
```

(*End definition for* `\marg`.)

`\oarg` `\oarg{text}` prints [⟨*text*⟩], 'optional argument'.

```
%\providecommand\oarg[1]{%
%  {\ttfamily[}\meta{#1}{\ttfamily]}}
```

(*End definition for* `\oarg`.)

`\parg` `\parg{te,xt}` prints (⟨*te,xt*⟩), 'picture mode argument'.

```
\providecommand\parg[1]{%
  {\ttfamily(}\meta{#1}{\ttfamily)}}
```

(*End definition for* `\parg`.)

B.6 DocInclude

```
%\@addtoreset{CodelineNo}{part}
\@addtoreset{CodelineNo}{chapter}
```

`\DocInclude` More or less exactly the same as `\include`, but uses `\DocInput` on a `dtx` file, not `\input` on a `tex` file. [6] [7] [8]

```
\def\partname{File}
```

[6] v2.0b,1994/03/14,Rename from `\docinclude`.
[7] v2.0f,1994/03/25,Use `\part`.
[8] v2.0u,1999/08/08,Also works for .fdd (M. Schroeder).

(*End definition for* `\DocInclude`.)

```
\def\task#1#2{}
\endinput

\newcommand*{\DocInclude}[1]{%
  \relax
  \clearpage
  \docincludeaux
  \IfFileExists{#1.fdd}{%
    \def\currentfile{#1.fdd}}{\def\currentfile{#1.dtx}
   }%
  \ifnum\@auxout=\@partaux
    \@latexerr{\string\include\space cannot be nested}\@eha
  \else \@docinclude#1 \fi}
%
\def\@docinclude#1 {\clearpage
  \if@filesw \immediate\write\@mainaux{\string\@input{#1.aux}}\fi
\@tempswatrue\if@partsw \@tempswafalse\edef\@tempb{#1}\@for
\@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswatrue\fi}\fi
\if@tempswa \let\@auxout\@partaux \if@filesw
\immediate\openout\@partaux #1.aux
\immediate\write\@partaux{\relax}\fi
```

We need to save (and later restore) various index-related commands which might be changed by the included file.

```
\let\@phddoc@PrintIndex\PrintIndex
\let\PrintIndex\relax
\let\@phddoc@PrintChanges\PrintChanges
\let\PrintChanges\relax
\let\@phddoc@theglossary\theglossary
\let\@phddoc@endtheglossary\endtheglossary
\part{\currentfile}%
  {\let\ttfamily\relax
  \xdef\filekey{\filekey, \thepart={\ttfamily\currentfile}}}%
\DocInput{\currentfile}%
\let\PrintIndex\@phddoc@PrintIndex
\let\PrintChanges\@phddoc@PrintChanges
\let\theglossary\@phddoc@theglossary
\let\endtheglossary\@phddoc@endtheglossary
\clearpage
\@writeckpt{#1}\if@filesw \immediate\closeout\@partaux \fi
\else\@nameuse{cp@#1}\fi\let\@auxout\@mainaux}
```

`\codeline@wrindex`

```
\gdef\codeline@wrindex#1{\if@filesw
        \immediate\write\@indexfile
            {\string\indexentry{#1}%
            {\filesep\number\c@CodelineNo}}\fi}%
```

(*End definition for* `\codeline@wrindex`.)

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

`\aalph` Special form of **`\alph`** as currently **`source2e.tex`** includes more than 26 files

[9].

```
\def\aalph#1{\@aalph{\csname c@#1\endcsname}}
\def\@aalph#1{%
  \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or
         j\or k\or l\or m\or n\or o\or p\or q\or r\or s\or
         t\or u\or v\or w\or x\or y\or z\or A\or B\or C\or
         D\or E\or F\or G\or H\or I\or J\or K\or L\or M\or
         N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or
         X\or Y\or Z\else\@ctrerr\fi}
```

(*End definition for* `\aalph`.)

`\docincludeaux` [10] [11]

```
\def\docincludeaux{%
  \def\thepart{\aalph{part}}\def\filesep{\thepart-}%
  \let\filekey\@gobble
% add to index prologue
  \g@addto@macro\index@prologue{%
    \gdef\@oddfoot{\parbox{\textwidth}{\strut\footnotesize
       \raggedright{\bfseries File Key:} \filekey}}%
    \let\@evenfoot\@oddfoot}%
  \global\let\docincludeaux\relax
%
 \gdef\@oddfoot{%
   \expandafter\ifx\csname ver@\currentfile\endcsname\relax
    File \thepart: {\ttfamily\currentfile} %
   \else
    \GetFileInfo{\currentfile}%
    File \thepart: {\ttfamily\filena
    me} %
    Date: \filedate\ %
    Version \fileversion
    \fi
    \hfill\thepage}%
% one sided paper
 \let\@evenfoot\@oddfoot}%
```

(*End definition for* `\docincludeaux`.)

```
\def\task#1#2{}

/class
```

[9] v2.0n,1994/05/27,Use uppercase letters, for makeindex.
[10] v2.06,1994/03/31,Use `\footnotesize` in file key..
[11] v2.0k,1994/05/21,Use `\aalph`.

Index

Symbols

\#(#) a772
\# 2002, 2544
\$($) a772
\%(%) a772
\% 466, 654, 1780, 2002, 2062, 2882, 2884, 2887, 2888, 2900, 2921
\&(&) a772
\(. 267
\) 267
* 267
* 1973, 1993, 2002
\- 35
\- 2002, 2526
.tex (file extension) a785
.tfm (file extension) a208
/errors/boolean expected 670
/errors/unknown choice value . . 670
/errors/unknown key 670
/errors/value forbidden 669
/errors/value required 669
/utils/exec 669
\: 267
\; 266
\@ 88
@@ commands:
\@@_doc_aux_env: 1463, 1477
\@@_doc_aux_env_star 1469, 1476
\@@_doc_aux_key: 1501, 1523
\@@_doc_aux_key_star 1512, 1522
\@@_doc_color_aux:n . 138, 153
\@@_doc_color_star:n 144, 152
\g_@@_macrocode_colorize_bool 8
\@@_print_docfile:n . 1705, 1715
\@@_document_types_clist . 881
\@@fileswith@pti@ns 519
\@@hyph 34
\@@italiccorr 35
\@@par 34
\@@unprocessedoptions 520
\@@warning 70
\@Alph 149
\@Esphack 83
\@M 80
\@Roman 149
\@activechar@info 453
\@addmarginpar 469
\@addtobot 439, 461
\@addtocurcol 439
\@addtodblcol 440
\@addtofilelist 102
\@addtonextcol 439, 466
\@addtoreset 148
\@addtotoporbot 439, 462
\@afterheading 909
\@alph 149
\@arabic 149
\@argdef 40
\@autoerr 71
\@autoload 55
\@backslashchar 49
\@badcrerr 73
\@badend 72
\@badlinearg 73
\@badmath 72
\@badpoptabs 72
\@badrequireerror 521
\@badtab 72
\@begindocumenthook 519
\@begindvi 456
\@beginparpenalty 288
\@bibitem 404
\@biblabel 406
\@bitor 437
\@boxfpsbit 474
\@break@tfor 62
\@bsphack 82, 83
\@car 39
\@cdr 39
\@cflb 457
\@changed@cmd 112
\@changed@x 112

\@charlb 100
\@charrb 100
\@check@c 47
\@check@eq 48
\@cite 406
\@cite@ofmt 406
\@citex 404
\@classoptionslist 508
\@cls@pkg 520
\@clsextension 508
\@clubpenalty 94
\@combinedblfloats 458
\@combinefloats 435, 456
\@comdblflelt 458
\@comflelt 457
\@cons 38
\@ctrerr 71
\@currentlabel 250
\@currenvir 253
\@currext 508
\@currname 508
\@currnamestack 508
\@dblarg 54
\@dblflset 372
\@dblfpbot 483
\@dblfpsep 483
\@dblfptop 483
\@dec@text@cmd 112
\@declaredoptions 507
\@definecounter 148
\@depth 35
\@dischyph 35
\@doclearpage 448
\@documentclasshook 507
\@doendpe 294
\@dofilelist 103
\@donoparitem 295
\@eha 70
\@ehb 70
\@ehc 70
\@ehd 70
\@empty 61
\@emptycol 445
\@enddocumenthook 519
\@endfloatbox 376
\@endparenv 294
\@endparpenalty 288
\@endpelttrue 295
\@ensuredmath 272
\@enumdepth 299
\@eqcnt 270
\@eqncr 271
\@eqnnum 269, 272
\@eqnsel 270
\@eqpen 270
\@esphack 83
\@expandtwoargs 48
\@filelist 102
\@fileswithoptions 516
\@firstofone 48
\@firstoftwo 48
\@flcheckspace 476
\@float 372
\@flsetnum 475
\@flsettextmin 476
\@flstop 473
\@fltovf 73
\@flupdates 477
\@fnsymbol 150
\@footnotetext 383
\@for 61
\@forloop 61
\@fornoop 61
\@fpbot 482
\@fpsep 482
\@fpstype 473
\@fptop 482
\@getfpsbit 474
\@getpen 81
\@gnewline 81
\@gobble 48
\@gobblefour 48
\@gobbletwo 48
\@height 35
\@hspace 88
\@hspacer 88
\@iden 48
\@if 46
\@ifclasslater 509
\@ifclassloaded 509
\@ifclasswith 510
\@ifdefinable 43
\@iffileonpath 100
\@ifnch 53
\@ifnextchar 52
\@iforloop 61
\@ifpackagelater 509
\@ifpackageloaded 509
\@ifpackagewith 510
\@ifstar 54
\@ifundefined 52
\@iinput 101
\@include 98
\@inlabel 288
\@inmatherr 74
\@inmathwarn 112
\@input 101
\@input@ 101

\@item 295
\@itemdepth 299
\@itemitem 300
\@itemlabel 289
\@itempenalty 288
\@itemsep 283
\@iwhiledim 60
\@iwhilenum 60
\@iwhilesw 61
\@ixpt 175
\@kludgeins 471
\@labels 288
\@latex@error 69
\@latex@info 69
\@latex@info@no@line 69
\@latex@warning 69
\@latex@warning@no@line . . . 69
\@latexbug 73
\@latexerr 70
\@lbibitem 404
\@listdepth 288
\@loadwithoptions 515
\@mainaux 93
\@makecol 435, 450
\@makefcolumn 434
\@makefnmark 590
\@makefntext 590
\@makespecialcolbox 451
\@maxdepth 442
\@minus 35
\@missingfileerror 102
\@mklab 295
\@multiplelabels 249
\@namedef 37
\@nameuse 38
\@needsformat 516
\@newcommand 40
\@newenv 45
\@newenva 44
\@newenvb 44
\@newl@bel 249
\@newline 81
\@newlist 289
\@next 440
\@nnil 61
\@no@lnbk 79
\@no@pgbk 79
\@nocnterr 71
\@nocounterr 71
\@nodocument 72
\@noitemerr 74
\@nolnerr 71
\@normalcr 80
\@notdefinable 71
\@notprerr 74
\@obsoletefile 102
\@onelevel@sanitize 54
\@onlypreamble 39
\@opcol 435, 449
\@options 514
\@outputpage 434, 453, 454
\@parmoderr 73
\@partaux 94
\@pkgextension 508
\@plus 36
\@popfilename 508
\@preamblecmds 39
\@preamerr 73
\@protected@testopt 42
\@providesfile 511
\@ptionlist 509
\@ptsize 567
\@pushfilename 508
\@qend 52
\@qrelax 52
\@rc@ifdefinable 43
\@reargdef 43
\@refundefined 248
\@reinserts 451
\@removeelement 62
\@reqcolroom 473
\@reset@ptions 519
\@resetactivechars 453
\@resethfps 475
\@roman 149
\@sanitize 54
\@savsf 82
\@savsk 82
\@scolelt 459
\@sdblcolelt 459
\@seccntformat 910
\@secondoftwo 48
@sect commands:
\@sect: 41, 42, 53
\@sect: 905
\@seqncr 271
\@setckpt 100
\@setfpsbit 475
\@setref 248
\@settodim 154
\@settopoint 154
\@slowromancap 149
\@spaces 70
\@specialoutput 447
\@specialpage 455
\@sptoken 53
\@sqrt 270
\@ssect 907

\@star@or@long 40
\@startcolumn 435, 458
\@startdblcolumn 435
\@stpelt 147
@tempa commands:
\@tempa: 185, 1652
\@tempboxa 154
\@testfp 474
\@testopt 41
\@textbottom 453
\@textfloatsheight 473
\@texttop 453
\@tfor 61
\@thirdofthree 48
\@toodeep 72
\@topnewpage 445
\@topsep 287
\@topsepadd 283, 287
\@totalleftmargin 288
\@trivlist 292
\@tryfcolumn 434, 459
\@twoclasseserror 521
\@twoloadclasserror 521
\@typeset@protect 51
\@unexpandable@noexpand ... 50
\@unexpandable@protect 50
\@unknownoptionerror 520
\@unusedoptionlist 508
\@use@ption 514
\@viiipt 175
\@viipt 174
\@vipt 174
\@vpt 174
\@vtryfc 459
\@warning 70
\@wckptelt 99
\@whiledim 60
\@whilenoop 60
\@whilenum 60
\@whilesw 60
\@whileswnoop 60
\@width 35
\@writeckpt 99
\@wtryfc 460
\@x@protect 51
\@xaddvskip 84
\@xargdef 40
\@xdblarg 54
\@xeqncr 271
\@xfootnotemark 384
\@xifnch 54
\@xiipt 175
\@xipt 175
\@xivpt 175
\@xnewline 80
\@xpt 175
\@xsect:n 908
\@xtryfc 460
\@xviipt 175
\@xxpt 175
\@xxvpt 175
\@yargd@f 42
\@yargdef 42
\@yeqncr 271
\@ytryfc 460
\@ztryfc 461
\[..... 268, 273
\{ 160
\addtolength 153
\newlength 153
\setlength 153
\settodepth 153
\settoheight 153
\settowidth 153
\} 160
\] 268, 273
\^ .. 170, 686, 688, 689, 2002, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2130, 2637, 2875, 2876, 2878, 2883, 2922
\= (¯) 780
\{ ({) 772
\} (}) 772
m 765
_(_) a772
_meaning 1751

\␣ 157, 236, 1183
\if@newlist 288
\if@noparitemfalse 289
\if@noparitemtrue 289
\set_borderline_aux:nn ... 894

A

\a 1
\AA 1
\aa 1
\AAA 1789, 1798
\aalph 211, 220, *685*
\abovedisplayskip 274
\abovedisplayskip *416*
\abstractname 11, 2354
\ABX 1929
accents 778, 779
acute (´) 778
arc (˘) 778

breve (˘) 778
caron (a˘) 778
\active@math@prime 267
\actualchar 1517, 1522, 2386, 2388, 2433, 2486, 2813, 2815
add commands:
\add_module_prefix:n 1796, 1801, 1804, 1807
add_to_callback 1
\add@accent 115
\add_to_callback *544*
\addcontentsline 397
\addcontentsline . 1, 72, 77, 1620, 1628, 2123, 2297, 2528, 2631, 2742
\addpenalty 85
\addpenalty 23
\AddPrefix 1802
\addto@hook 174
\addtocontents 1
\addtocounter 147
\addtolength 153
\addtolength 149, 150, 151
\addvspace 85
\addvspace . 24, *78*, *84*, 3027, 3142, 3150
\adp . . . 1935, 1942, 1945, 1946, 1947
\adsorbate 2161
\adsorbent 2162
\aegean 141
after commands:
\after_block:n 208
\afterassignment 2707
\aht 1935, 1941, 1944, 1947
\alldoztallies 2063
\allepsdice . . . 2018, 2019, 2020, 2021, 2022, 2023, 2033
\allocationnumber 1
\allowbreak 1, 914
\Alph 149
\alph 148
\alphabet 193
\alphabet *193*
\AlphAlph 361
\alphalph 354, 1697
\AlsoImplementation 1594
\alsoname 2370
\AltMacroFont 2557
\and *357*
\and 1
animals 768
\antimuon 2163
\antineutrino 2164
\antineutron 2165
\antiproton 2096, 2166
\antiquark 2167
\antiquarkb 2168
\antiquarkc 2169
\antiquarkd 2170
\antiquarks 2171
\antiquarkt 2172
\antiquarku 2173
\anyon 2174
\appendix 583
\appendixname 2357
\arabic 148
\arccos(arccos) a783
\arcsin(arcsin) a783
\arctan(arctan) a783
\Arg 447, *804*, *822*
\arg(arg) a783
arg commands:
\X_opt_arg 2440, 2447
arguments (environment) *806*
\arial 2834
Arial (font) 799
Arial Unicode MS (font) 204
\ARR 2214
\arrayrulecolor 1114, 1126
\arraystretch 1007
arrows 768
\ARRtrue 2216
asdf 727
\assumption 2175
\AtBeginDocument 520
\AtBeginDocument 139, 1145
\AtBeginDvi 442
\AtEndDocument 520
\AtEndDocument 2219, 2289, 2316, 2538
\AtEndOfClass 520
\AtEndOfPackage 520
\AtNextShipout 425
\AtNextShipout *425*
\atom 2176
atomic math objects 783
\attribute *541*
\aut@global 55
\author *357*
\author 1
auto 886
\auxm 3146, 3160
\awd 1935, 1940, 1943

B

background (package) 427
\backmatter 574
BAR 602

Basic Multilingual Plane 203
\begin 255
\begin 8, 17, 43, *255*, *274*, *299*, *328*, *335*, *499*, *589*, *728*, *730*, *731*, *732*, *741*, *744*, *746*, *754*, *755*, *756*, *789*, *797*, *798*
\belowdisplayskip . . . 3142, 3145
\bezier 1
\bfseries . . 48, 135, 194, 225, 316, 339, 393, 457, 488, 515, 859, 940, 983, 985, 986, 1207, 1224, 1248, 1483, 1531, 1707, 1737, 1915, 1955, 2473, 2493, 2509, 2535, 3021, 3024, 3042, 3046, 3050, 3058, 3154, 3175
\bgroup 1
\bibcite 404
\bibdata 405
\bibitem 404
\bibliography 405
bibliography font 205
\bibliographystyle 405
\bibliographystyle *403*
\bibname 13, 2355
\bibstyle 405
\Big . 2263, 2269, 2282, 2287, 2290, 2291, 2292, 2294
\big 263
\big 2453
\Bigassumption 2178
\bigassumption 2177
\bigbreak 1
\Bigg 2453
\biggassumption 2179
\Biggl 3216
\biggl 3216
\Biggm 3211
\biggm 3211
\Biggr 3222
\biggr 3222
\Bigl 3216
\bigl 3216
\Bigm 3211
\bigm 3211
\Bigr 3222
\bigr 3222
\bigskip 86
\bigskip 909, 1554, 1579, 1593, 2746
\bigskipamount 87
\blackacc . . 1767, 1937, 1951, 2331, 2349, 2426, 2430
\blackacchack . 1938, 1950, 2298, 2304, 2346, 2353, 2357, 2361, 2370
\blackacctwo 1952, 2437
\Bmesonminus 2180
\Bmesonnull 2181
\Bmesonplus 2182
\bmod 262
\bmod *262*
bobhook (package) 424, 425
\boldmath *166*
\bond 2183
book
class 566
environments
thebibliography 589
bool commands:
\bool_gset_false . . 24, 36, 43, 47, 51, 53, 57, 262, 264, 266, 268, 314, 315, 316, 323, 324, 325, 332, 333, 334, 340, 341, 342, 353, 354, 355, 362, 363, 364, 371, 372, 373, 379, 380, 381, 463, 465, 1595, 1980
\bool_gset_true . 23, 30, 31, 35, 39, 41, 45, 46, 49, 55, 89, 94, 98, 99, 103, 107, 141, 280, 281, 282, 283, 284, 459, 461, 1598
\bool_if 16, 21, 22, 110, 143, 166, 201, 204, 295, 310, 310, 318, 327, 336, 349, 357, 366, 375, 468, 472, 475, 479, 482, 487, 553, 571, 584, 679, 711, 722, 765, 771, 785, 787, 816, 965, 1020, 1023, 1039, 1040, 1054, 1059, 1071, 1076, 1109, 1128, 1292, 1294, 1300, 1330, 1353, 1431, 1432, 1474, 1602, 2092, 2106, 2128, 2236, 2363
\bool_lazy_all 1419
\bool_lazy_and . 1085, 2170
\bool_new 10, 11, 12, 13, 13, 14, 15, 16, 17, 18, 19, 22, 23, 26, 30, 31, 32, 33, 34, 35, 36, 37, 41, 42, 43, 44, 45, 53, 59, 71, 72, 73, 74, 75, 81, 98, 141, 261, 263, 265, 267, 1979
\bool_set 951
\bool_set_false 469, 547, 831, 836, 845, 924, 925, 926, 927, 928, 929, 930, 1168, 1180, 1185, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 2076, 2086, 2447
\bool_set_true . . 14, 76, 77, 476, 548, 825, 830, 837, 842, 843, 844, 850, 851, 860, 886,

887, 931, 1147, 1148, 1153, 1154, 1158, 1161, 1165, 1166, 1167, 1173, 1174, 1179, 1186, 1190, 1360, 2127, 2446
\bool_show 386
\c_false_bool 250, 769, 772, 1285, 2326
\c_true_bool 249, 1274
\boolcolor 1822
border-top-width 886
\bordermatrix 266
\boseDistrib 2184
\boson 2185
\botfigrule 483
\BOTLIST *435*
\botmark *420*
\bottomfraction 480
\bottomfraction *368*
\bottomrule 2374
\box 417
\box 2963
box commands:
\box_clear 1227
\box_gclear 921
\box_if_empty 918, 1104
\box_new 21, 65, 66
\box_use_drop ... 964, 1132
\box_wd 950
\box255 ..... *423*, *425*
\boxcolor 1835
\brace 264
\bracescolor 1911
\brack 263
\bracketscolor 1915
\break 22
\BreakableUnderscore 560
bricks (pattern) *791*
\bslash 2323, 2427
\buildrel 265

C

\c@bottomnumber 480
\c@dbltopnumber 481
c@enumi counter 299
c@enumii counter 299
c@enumiii counter 299
c@enumiv counter 299
\c@equation 269
c@equation counter 269
\c@errorcontextlines 70
\c@topnumber 480
\c@totalnumber 480
calc (package) 155
call commands:
call_callback 1
\call_callback *545*
callback commands:
callback_descriptions .. 1
\callback_descriptions ... *545*
\caption 371
\caption 2348, 2674
\captionsenglish 2376
\CaptionsList 2342
\carrow 2796
case 886
\cases 265
catcode commands:
\c_catcode_active_space_tl 180
\c_catcode_other_space_tl 749, 1341
\catcodetable *541*
\catcodetable@atletter ... *542*
\catcodetable@initex *542*
\catcodetable@latex *542*
\catcodetable@string *542*
\catcoding *111*
\cB 1756, 1768, 1781, 1874, 1883, 1904, 1923, 1934, 1966, 1976, 2009
\ccname 2366
\ccolor 1832
\cctabcolor 1841
\cE 1756, 1768, 1781, 1874, 1885, 1904, 1923, 1934, 1966, 1976, 2009
center (environment) 1
\centering 1, 324, 349, 2673
\centerline 1
\ch@ck 17
\changelogfont 3048
\changeloglabelfont 3050
\changes *843*
\chapter 912
\chapter 2, 3, 6, 347
chapter commands:
\chapter_open_any_bool 354, 363, 366, 381
\chapter_open_anywhere_bool 355, 364, 373, 375
\chapter_open_left_bool 349, 362, 371, 379
\chapter_open_right_bool 353, 357, 372, 380
\chaptername 885
\chaptername 17, 188, 2356
\chapterparent 890
\char .. 60, 154, 157, 160, 448, 1024, 1038, 1041, 1325, 1529, 1834,

1889, 2097, 2705, 2714, 2736, 2847, 2888, 3056
char commands:
\char_set_active_eq .. 934
\char_set_catcode 614
\char_set_catcode_active .. 170, 935, 2058, 2059, 2077, 2078
\char_set_catcode_letter 397, 398, 399
\char_set_catcode_other 2057, 2060, 2061
\char_set_lccode ... 2062, 2063, 2064, 2065, 2066
character 193
\charcolor 1819
\chardef@text@cmd 112
check marks 782
\check@command 47
\CheckCommand 47
\CheckEncodingSubset 137
\checkmark(✓) a782
\choose 263
\circle 1
\circledR(®) a782
\citation 404
\cite 404
\cite *403*
CJK 208
CJKV 208
\cl@@ckpt 148
\cl@page 245
class environment 789
\class 2728
\ClassError 67
\ClassError 1, 2284
classes
fontdef.dtx 262
kernel-ltmath.dtx 262
KOMA 567
KOMA 210
ltxdoc 390, 726, 746
memoir 210
Octavo 567
octavo 485
phddoc 390, 681, 721, 726
standard 566
\ClassInfo 67
\ClassWarning 67
\ClassWarningNoLine 67
\cleardoublepage 444
\cleardoublepage . 280, 320, 359
\clearpage 435, 444
\clearpage .. *94*, *98*, 173, 182, 203, *433*, *434*
\cleartoevenpage 311, 350
clist commands:
\clist_clear 932
\clist_count:N *810*
\clist_count:n *810*
\clist_gput_right 74
\clist_gset 79, 120, 130, 1729, 1761, 1889
\clist_map_function 1632, 1784, 1895, 2245, 2264
\clist_map_inline 114, 125, 179, 186, 415, 420, 492, 1061, 1625, 1732, 1738
\clist_new 10, 68, 75, 80, 111, 119, 129, 1786
\clist_put_right . 115, 1627
\clist_set 859, 2380
\g_docinput_clist . 10, *807*, *851*, 1627, 1632
\clistcolor 1817
\closeout 204, 1680
\cls 156, 452, *804*
\CLSLpig 2200
\CLSLpipe 1907, 1913, 1919
\clubpenalty *94*, 223, 229, 232, 236
\cmd .. 153, 420, 442, *803*, *812*, *825*, 1757, 2608, 2609, 2653, 3054
\cmdI . 1758, 1766, 1767, 2653, 2655, 2657
\cmdIp 1768, 2656
\cmdW 1767, 2656
\cmdX 1766, 2654
\cnt 3062
code2001 203
codecolorize 859
\CodedocExplain 512
\CodedocExplainEXP 512
\CodedocExplainREXP 512
\CodedocExplainTF 512
\codeexample 2908
\codeexamplebox 2907, 2929, 2940, 2963
\codeexamplenl 3000, 3011
\codeexamplewidth .. 2891, 2905, 2911, 2940, 2955, 2960
\codelinefont 1736
\CodelineIndex 145
CodelineNo 401
\CodelineNumbered 146
\codetwothousand 203
\codetwothousandone 203

coffin commands:
\coffin_clear 920, 922, 923
\coffin_join . 967, 971, 975, 984, 988, 992
\coffin_new ... 17, 18, 19, 20
\coffin_typeset .. 979, 996
\coffincolor 1834
coffinkeywords commands:
\coffinkeywords_clist 1761
\coffinkeywordsregex 1764
\col@number 443
\colOpt 1008, 3103, 3104
\color 14, 48, 103, 157, 208, 213, 214, 215, 222, 232, 239, 240, 241, 250, 262, 268, 269, 270, 279, 314, 315, 316, 322, 352, 360, 361, 362, 363, 364, 393, 401, 402, 403, 403, 404, 635, 651, 655, 663, 665, 680, 681, 709, 714, 718, 722, 726, 729, 736, 737, 738, 739, 803, 807, 812, 859, 884, 888, 893, 940, 978, 983, 984, 985, 986, 1023, 1031, 1039, 1156, 1534, 1703, 1707, 1738, 1746, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1867, 1915, 1955, 2360, 2558, 2708, 2936, 3024, 3154, 3175
\colorbox 2936, 2973
\colorizebraces 1916, 2016
\colorizebrackets .. 1907, 2018
\colorizecomments .. 2005, 2047
colorizecomments 1981
\colorizehandlers ... 1777, 2021
\colorizeifs 1752, 2046
\colorizemathoperator ... 1959, 2019
\colorizemodulekeywords . 1784, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2032, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045
\colorizesquarebrackets . 1927, 2017
colorizesquarebrackets .. 1907
\colorizexparsedeclarations 1969
\colorlet 2892, 2893
Colors
thecs a726
themeta a746
theoption a797
\columnsep . 1602, 1605, 2751, 2754
\columnseprule 1601, 2751
combo commands:
\combo_if_bool 141, 143
\command 2711
\comment 470, 477, *823*
\commenthandler 2884, 2900, 2933
\commentshack 2048
Computer Modern (font) 192
\conductivity 2186
\contentsline 1
\contentsname 591
\contentsname 2358
\continueLineNumber *731*
\continuelinenumber 731
\continuelinenumber 329
\copyright 214
\cos(cos) a783
\cosh(cosh) a783
\cot(cot) a783
\coth(coth) a783
Counters
c@enumi 299
c@enumii 299
c@enumiii 299
c@enumiv 299
c@equation 269
foocounter 747
incsyms 772, 779
totalsymbols 770
\covbond 2187
\create_callback *545*
\CS *37*
\cs 104, 153, 420, 443, 519, *803*, *803*, *803*, *807*, *812*, *825*, *842*, 2204, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 3054
cs commands:
\cs:w 17, 25, 28, 31, 34, 37, 39, 41, 44, 69, 70, 73, 76, 80, 91, 103, 112, 117, 118, 127, 146, 153, 156, 168, 170, 176, 187, 189, 261, 262, 273, 280, 281, 295, 296, 308, 309, 310, 315, 316, 317, 322, 323, 324, 327, 329, 330, 331, 336, 337, 338, 343, 344, 345, 352, 353, 354, 355,

360, 361, 362, 367, 368, 369, 374, 375, 376, 381, 382, 383, 388, 389, 390, 395, 396, 397
\cs_end 17, 25, 28, 31, 34, 37, 39, 41, 44, 70, 71, 73, 76, 82, 91, 103, 112, 119, 120, 129, 146, 153, 156, 168, 170, 178, 187, 189, 261, 262, 273, 280, 281, 295, 296, 308, 309, 310, 315, 316, 317, 322, 323, 324, 327, 329, 330, 331, 336, 337, 338, 343, 344, 345, 352, 353, 354, 355, 360, 361, 362, 367, 368, 369, 374, 375, 376, 381, 382, 383, 388, 389, 390, 395, 396, 397
\cs_generate_variant . 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 117, 123, 261, 606, 675, 775, 782, 1029, 1035, 1082, 1281, 1287, 1351, 1409, 1784, 1785, 1801, 2341, 2368
\cs_gset 2, 6, 216, 244, 259, 263, 276, 306, 309, 313, 320, 327, 334, 341, 348, 351, 358, 365, 372, 379, 386, 393, 395, 400, 405, 406, 1529, 1644, 1670, 1686, 1695, 1702, 1713, 1869, 1878, 2005, 2014, 2291
\cs_gset_eq ... 48, 407, 408, 1026, 1712, 1753, 1754, 1755, 1906
\cs_gset_protected . 2303, 2317
\cs_if_exist 115, 1000, 1333, 1715
\cs_if_exist:NTF . *804*, *805*
\cs_if_exist_p:N . *804*, *805*
\cs_if_exist_use 4, 5, 6, 7, 8, 68, 79, 160, 161, 162, 163, 164, 182, 183, 184, 185, 186, 215
\cs_if_free 1888
\cs_new ... 11, 15, 21, 78, 100, 110, 112, 162, 192, 198, 214, 216, 232, 234, 240, 242, 246, 253, 256, 272, 292, 296, 321, 327, 339, 346, 360, 361, 415, 435, 438, 800, 949, 984, 1083, 1205, 1796, 1844, 1865, 1876, 1907, 1932, 1933, 1969, 2542, 2546, 2556
\cs_new_eq 86, 87
\cs_new_nopar 70
\cs_new_protected 118, 134, 142, 164, 171, 262, 303, 308, 545, 607, 621, 625, 641, 644, 647, 660, 668, 676, 696, 703, 709, 727, 729, 739, 744, 763, 776, 801, 866, 876, 889, 901, 907, 911, 916, 937, 941, 945, 954, 959, 961, 1001, 1017, 1030, 1036, 1042, 1057, 1069, 1102, 1122, 1199, 1211, 1239, 1243, 1252, 1254, 1272, 1282, 1288, 1297, 1313, 1324, 1346, 1352, 1358, 1371, 1395, 1402, 1410, 1417, 1436, 1441, 1448, 1463, 1615, 2079, 2141, 2168, 2220, 2234, 2329, 2344, 2369, 2397, 2431, 2439, 2463, 2468, 2475, 2480, 2489
\cs_new_protected_nopar 940
\cs_set . 8, 8, 9, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 52, 53, 63, 89, 96, 108, 120, 131, 133, 138, 142, 144, 157, 172, 176, 177, 179, 206, 208, 222, 253, 256, 257, 258, 259, 260, 266, 283, 307, 314, 335, 378, 458, 505, 547, 591, 637, 680, 724, 767, 811, 853, 901, 917, 933, 1007, 1008, 1029, 1035, 1044, 1048, 1090, 1176, 1189, 1222, 1246, 1258, 1260, 1271, 1288, 1302, 1323, 1342, 1350, 1450, 1453, 1455, 1463, 1469, 1481, 1486, 1528, 1534, 1548, 1554, 1559, 1570, 1571, 1581, 1582, 1591, 1592, 1595, 1598, 1610, 1613, 1614, 1616, 1617, 1625, 1626, 1633, 1638, 1639, 1642, 1651, 1705, 1712, 1719, 1752, 1777, 1787, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1867, 1916, 1927, 1937, 1959, 1969, 1974, 1990, 1996, 2007, 2234, 2241, 2242, 2247, 2259, 2263, 2265, 2266, 2296, 2302, 2328, 2376, 2382, 2456, 2470
\cs_set_eq 18, 419, 442, 443, 444, 445, 1226, 1658, 1661, 1662, 1663, 1664, 1665, 1666,

1669, 1674, 1675, 1676, 1677, 1684, 1699, 1710, 1725
\cs_set_protected 244, 409, 2070
\csc(csc) a783
\cscolor 1813
\csname 648
\csname 1, *37*, 211, 230
\ctan 785
\ctan 2533
\curr@fontshape 168
currency symbols 765, 788
\currentfile . . 176, 195, 197, 198, 230, 231, 233, 1638, 1639, 1667, 1671, 1673, 1715, 1716, 1718
\CurrentOption 508
\CurrentOption 500
\CurrentOption 65, *518*
\currentsectionname 36, 146, 148
\cvaref 1007, 3101
\cxset 1, 5, 60, 63, 85, 114, 122, 142, 149, 150, 160, 162, 178, 186, 187, 195, 200, 214, 221, 238, 239, 262, 265, 265, 268, 271, 274, 278, 281, 292, 299, 300, 309, 334, 351, 359, 377, 394, 420, 437, 474, 507, 521, 564, 608, 654, 697, 723, 741, 783, 827, 869, 897, 1020, 1047, 1054, 1064, 1071, 1073, 1080, 1090, 1103, 1106, 1112, 1126, 1128, 1133, 1134, 1161, 1171, 1190, 1220, 1250, 1288, 1317, 1349, 1351, 1379, 1391, 1393, 1402, 1427, 1435, 1477, 1479, 1508, 1552, 1553, 1581, 1612, 1637, 1640, 1668, 1670, 1693, 1694, 1705, 1734, 1761, 1788, 1816, 1907, 1934, 1962, 1999, 2038, 2066, 2094, 2131, 2173, 2175, 2201, 2263, 2291, 2315, 2316, 2344
cypriote (script) 764

D

\dashbox 1
\date *357*
\date 1, *7*
\dateitalian 2373
\dateukrainian *1*
\day 144, 172, 217, 260, 325, 326, 373, 416, 456, 499, 545, 586, 631, 632, 676, 719, 763, 806, 849, 892, 1043, 1086, 1128, 1155, 1183, 1212, 1243, 1285, 1311, 1346, 1374, 1432, 1459, 1502, 1545, 1547, 1575, 1605, 1634, 1662, 1703, 1727, 1756, 1783, 1810, 1838, 1929, 1956, 1997, 2036, 2060, 2088, 2116, 2155, 2197, 2265, 2281, 2284, 2338
\dbend 1529, 1535
\dblfigrule 483
\dblfloatpagefraction 481
\dblfloatpagefraction *369*
\dblfloatsep 482
\dbltextfloatsep 482
\dbltopfraction 481
\dbltopfraction *368*
\dcircle 789, 798, 878
\dday 2260
\deadcycles *416*
\DEBUGOFF 37
\DEBUGON 17, 57
\declare . . 2707, 2724, 2736, 2847, 2853
\declare@robustcommand 50
\DeclareDocumentCommand . . . 2, 7, 121, 149, 179, 291, 331, 420, 423, 426, 432, 1382, 1392, 1393, 1519, 1563, 1573, 1583, 1594, 1597, 1600, 1612, 1623, 1631, 1633, 1727, 3226, 3230
\DeclareDocumentEnvironment 480, 491, 494
\DeclareEncoding *111*
\DeclareEncodingSubset 136
\DeclareEncodingSubset . . . *136*
\DeclareErrorFont 170
\DeclareExpandableDocumentCommand 416, 434, 436
\DeclareFixedFont 159
\DeclareFontEncoding 161
\DeclareFontEncodingDefaults 163
\DeclareFontFamily 160
\DeclareFontFamily 141, *160*
\DeclareFontShape 158
\DeclareFontShape 140, 142, 143, 144
\DeclareFontSubstitution . 162
\DeclareMathSizes 163
\DeclareMathSymbol . 2091, 2095, 2101
\DeclareOption 512
\DeclareOption 500

\DeclareOption 25, 28, 33, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 61, 63, 65
\DeclareOption* 512
\DeclareOption* 500
\DeclareOptionX 643
\DeclareOptionX* 643
\DeclarePreloadSizes 163
\DeclarePreloadSizes . . 18, 20, 21, 22, 25, 26, 27, 28, 29, 30, 34, 38, 44, 46
\DeclareProtectedCommand *91*, *91*
\DeclareRobustCommand 50
\DeclareRobustCommand 156, 452, 453, 455, 456, 1023, 1745, 2032, 2033, 2061, 2062, 2063, 2320, 2533, 2550, 2551, 2552, 3166
\DeclareSymbolFont 2090, 2093, 2100
\DeclareTextAccent 115
\DeclareTextCommand 112
\DeclareTextCommandDefault 114
\DeclareTextFontCommand . . 240
\DeclareTextSymbol 112
\def 153, 154, 167, 168, 176, 182, 211, 212, 219, 220, 242
\default@ds 512
\defaultnumberingsystem . . . 37
\defaultscriptratio 1
\defaultscriptscriptratio . . 1
\define@newfont 168
\definecolor 10, 11
\deflength 3071
\deg(deg) a783
\depthof 155
\DescribeOption 1486
\description 1545, 1562
\descriptionlabel 580
\det(det) a783
\detokenize . 60, 1024, 1717, 1899, 1900, 3091
\dim(dim) a783
dim commands:
\dim_compare_p 952
\dim_gzero_new 164, 165, 166, 167, 169, 170, 171, 172, 174, 175, 176, 177
\dim_if_exist 13
\dim_new 24, 1101
\dim_set 949, 1106
\dim_zero 947
\dimcolor 1818
\ding *202*, 645, 789, 798, 878, 1539, 2400, 2401, 2402, 2409, 2410
\dipole 2188
disable commands:
disable_callback 1
\disable_callback *545*
\DisableDocumentation 458
\DisableImplementation . . . 458
\displaylines 266
displaymath (environment) 1
\displaystyle . 2380, 2386, 3207
\displayverbfont 18
\Dmesonminus 2189
\Dmesonnull 2190
\Dmesonplus 2191
\do 9, 185
\doc 1998
doc commands:
\g_doc_functions_seq . . 11, 1032, 2179, 2195
\g_doc_macros_seq 12, 1303, 2181, 2193
doc left 749
doc right 749
\docAuxCommand 726
\docAuxCommand 121
\docAuxEnv 757
\docAuxEnv 1473
\docAuxKey 758
\docAuxKey 1519
\docAuxListingsStyle 291
\docClass 784
\docClass 2508
\docColor 726
\docColor 149
\docColorBox 1315
docCommand environment 746
docCommands environment 798
\docCounter 747
\docCounter 1048
\docCounter* 747
docEnv* 756
\docExtension 1719
\docextension 1724
\docFile 763
\docFile 1712
\docFileExtension 2526
\docfileextension 785
\docfileextension . . 2520, 2526
\docFilename 2519
\docfilename 785
\docfilename 2514, 2519
\docFont 799
\docFont 3168
\docFontArial *799*
\DocInclude 167, 171, 1633

\docinclude 683
\docincludeaux 174, 219, 1636, 1695
\DocInput 807
\DocInput 198, 1622, 1673
\DocInputAgain 1631
docKey environment 756
\docSymbol 787
\docSymbol 2660
\document 94
documentation (environment) . 466
\documentclass 514
\documentstyle 514
\docValue 727
\docValue 179, 3135
\docZ 783
\docZ 2456
\DoNotIndex 499, 1292, 1299
\dospecials 259
\dotfill 25
\doublecovbond 2192
\downarrow 2224
\DOZ 2053
\doz . 2061, 2062, 2064, 2065, 2066, 2067, 2068, 2069
\DOZe 2062
\DOZtrue 2055
\DOZx 2061
\dp 2558
draftwatermark (package) 425
\draw 22, 42, 2772, 2776, 2788, 2792, 2802, 2848, 2853
\DTMsetstyle 1

E

\edef 184, 420
\EDICE 2013, 2027
\EDICEtrue 2029
\egroup 1
\eject 1
\electron 2193
element commands:
\element_delimiter 110, 115
\else 180, 205, 218, 232
else commands:
\else: 181, 2254
\emph 315, 316, 1439
\emphasis 334
\emphasize 870
\empty 20
\empty 865
\EmptyPageContents ... 267, 271
\emptypagecontents 263, 268, 282
\EnableCrossrefs 147, 1638
\EnableDocumentation 458
\EnableImplementation 458
\encapchar .. 115, 1310, 1316, 1510, 1511, 1518, 1523, 1717, 1722, 1747, 1800, 1805, 1843, 1850, 1893, 1895, 1900, 1902, 2393, 2433, 2482, 2483, 2487, 2516, 2522, 2523, 2808, 2809, 2812, 2814, 2815, 2828, 2829, 2832, 2833, 3167
\encfive 2249, 2370, 2544
\encfour .. 2353, 2357, 2361, 2362, 2543
\encgreek 2250, 2251, 2545
\enclname 2365
encoding schemes 194
OML 194
OMS 194
OMX 194
OT1 194
U 194
encodings 128
\encone ... 2237, 2238, 2245, 2246, 2349, 2542
\end 1, 4, 23, 48, 274, 299, 328, 335, 416, 522, 589, 728, 730, 731, 732, 741, 744, 746, 754, 755, 756, 789, 797, 798
end commands:
\end: 708
\end@float 375
\endclass 2733
\endcodeexample 2999
\endcommand 2712
\endcomment 472, 479, 823
\endcsname 1, 211, 230
\enddescription 1549, 1575
\enddispListing 1172
\enddocument 253
\enddocument 249
\endenumerate 3246
\endfilecontents 521
\endfirsthead 2156
\endfoot 2159
\endgraf 20
\endgraf 505, 1534
\endgroup 285
\endhead 2155
\endinput 169, 522
\endlastfoot 2160
\endline 1
\endlist 293
\endlist .. 2556, 2733, 3082, 3092
\endlrbox 1
\endluaexample 963

\endmacroarguments 2546
\endmacroaruments 2556
\endmarglist 3099
\endminipage 313
\endofcodeexample 2926
\endpicture 1
\endsmallverbatim 3133
\endtabbing 1587
\endtcblisting 950, 953, 956, 959, 994, 997
\endtexexample 963
\endtextoken 2712
\endtheglossary . 194, 202, 1666, 1677
\endtheindex 1613, 2761
\endtrivlist 294
\endVerbatim 408
\endverbatim 259
\endverbatim 406
\enlargethispage 6
\enquote 536, 3169
\enskip 89
\enspace 78, 89
\ensuremath 272
\ensuremath . 622, 627, 638, 2564, 3181
enumerate environment 299
\enumerate 3243
\env 452, *804*, 3058
Environments
class 789
docCommand 746
docCommands 798
docEnv* 756
docKey 756
enumerate 299
eqnarray 274
equation 274
function 807
itemize 299
keymarglist 797
lrbox 307
macro 807
phd@doc@head 755
phdd_manual_entry 754
picture 333
picture 328, 335
quotation 278
scriptexample 728
sverbatim 730
texexample 741, 744
teXX 731
teXXX 732
thebibliography 589
verse 580
environments
picture 328
environments:
arguments *806*
center 1
displaymath 1
documentation 466
equation 1
function 491, *804*
implementation 466
macro 491, *805*
math 1
sloppypar 1
syntax 497, *804*
texnote 503, *805*
thebibliography *403*
variable 480, *805*, *806*
\epsdice .. 2018, 2019, 2020, 2021, 2022, 2023, 2031, 2032, 2034, 2035, 2036, 2037, 2038, 2039
eqnarray environment 274
equation environment 274
equation (environment) 1
\errorstopmode *539*, 2283
\errorsym 2194
escapable characters 772
\EStruename 1956, 1957
\etameson 2195
\etamesonprime 2196
euro signs 765, 788
\eval *248*
\evensidemargin 151
\everydisplay 1, *416*
\everymath 167
everypage (package) 425, 427
\everypar *285*, *294*, *416*
everyshi (package) 424, 425
\EveryShipout 425
\EveryShipout *425*
\example 789
\example 2715
\exciton 2197
\ExecuteOptions 500, 514
\ExecuteOptions 66
\ExecuteOptionsX 643
\exp(exp) a783
exp commands:
\exp_after 121, 127, 128, 248, 354, 361, 793, 1249, 1794, 2255, 2473
\exp_args 138, 260, 298, 411, 413, 592, 618, 692, 716, 728, 809, 879, 1065, 1079, 1293,

1309, 1481, 2408, 2486, 2487, 2491, 2492, 2520
\exp_last_unbraced .. 799, 1334, 2448
\exp_not .. 120, 233, 274, 276, 278, 282, 288, 290, 294, 299, 317, 589, 1075, 1329, 1452, 1794, 2244, 2348, 2471
\exp_not:n *816*
\expandafter 1, 153, 230
\expcolor 1826
\experimentalsym 2161
\ExplFileDate 1720
\ExplFileVersion 1721
\ExplMakeTitle *680*
\ExplSyntaxOff ... 7, 8, 15, 19, 19, 20, 50, 51, 59, 61, 81, 82, 87, 94, 121, 126, 126, 128, 139, 155, 169, 185, 196, 214, 242, 257, 261, 264, 274, 291, 305, 307, 312, 319, 344, 345, 378, 403, 433, 433, 453, 482, 504, 509, 513, 533, 1005, 1009, 1027, 1048, 1072, 1209, 1220, 1321, 1327, 1358, 1376, 1389, 1416, 1461, 1479, 1525, 1552, 1568, 1579, 1589, 1593, 1643, 1725, 1744, 1756, 1808, 1908, 1934, 1988, 2005, 2012, 2052, 2240, 2248, 2279, 2314, 2342, 2383, 2394, 2404, 2449, 2464, 2670, 3248
\ExplSyntaxOn . 1, 1, 1, 3, 4, 5, 7, 9, 9, 20, 21, 42, 51, 52, 59, 62, 62, 72, 84, 88, 95, 97, 122, 127, 129, 140, 156, 171, 197, 215, 243, 258, 262, 275, 285, 295, 306, 308, 313, 320, 345, 346, 404, 434, 434, 460, 483, 506, 510, 999, 1006, 1025, 1028, 1053, 1175, 1210, 1221, 1322, 1328, 1359, 1377, 1390, 1424, 1462, 1480, 1526, 1553, 1569, 1580, 1590, 1594, 1704, 1726, 1730, 1752, 1787, 1905, 1930, 1967, 1989, 2006, 2055, 2233, 2241, 2258, 2295, 2327, 2375, 2396, 2439, 2455, 2659, 3240
\externalsym 2162
\extrachap 6
\extractclass 2731, 2735
\extrafloats 15

F

\faBitcoin(฿) a788
\faBtc(฿) a788
\faCny(¥) a788
faDollar(faDollar) a788
\faEur(€) a788
faEur(faEur) a788
faEuro(faEuro) a788
\faGbp(£) a788
\faIls(₪) a788
faIls(faIls) a788
\faInr(₹) a788
faInr(faInr) a788
\faJpy(¥) a788
faJpy(faJpy) a788
\fakeusepackage 2044, 2059
\faKrw(₩) a788
\fancyvrbtrue 3
\faRmb(¥) a788
faRouble(faRouble) a788
\faRub(₽) a788
faRub(faRub) a788
faRuble(faRuble) a788
faRupee(faRupee) a788
faShekel(faShekel) a788
faSheqel(faSheqel) a788
\faTry(₺) a788
faTry(faTry) a788
faTurkishLira(faTurkishLira) a788
\faUsd($) a788
faUsd(faUsd) a788
\faViacoin(ⱥ) a788
\faWon(₩) a788
\faYen(¥) a788
\fbox 308
\fboxrule 308
\fboxrule 146
\fboxsep 308
\fboxsep 145
\FC 1927
\FCbar 2361, 2362
\FCfalse 2355
\feat 2501
\Ferli() a779
\Fermi() a779
\fermiDistrib 2163
\fermion 2164
\feyn 2082
\fi 180, 183, 185, 188, 204, 205, 209, 218, 238
fi commands:
\fi: 14, 193, 204, 236, 280, 297, 349, 392, 472, 519, 562, 606, 652, 695, 739, 781, 825, 867,

915, 931, 947, 1062, 1104, 2256, 2279
\figurename 2362
\filbreak 1
\file 452, 804
file commands:
\g_file_curr_name_str 1400
\file_if_exist:nTF 75
\file_input:n 77
file extension
.def 106
file extensions
.ist a396, a748
.tex a785
.tfm a208
cls a485
\filecontents 521
\filedate 236
\filekey . 197, 221, 225, 1670, 1671, 1699, 1707
\filena 234
\filename 1719
files:
float.dtx a429
gind.ist a396
phddoc.cls a675
puenc.def a197
sample.ist a396
source2e.pdf a3
source2e a5
template_html.tpl a763
testfont.tex a207
\filesep 209, 210, 220, 1691, 1694, 1698
\fileversion 237
\fill 88
\fill 272, 276
\Finale 1594
\firstmark 420
flippdf (package) 425
\FLIST 434
\floatpagefraction 481
\floatpagefraction 368
\floatsep 481
\floatsep 435
\flushbottom 410
\fmtname 1, 515
\fmtversion 1, 515
\fnsymbol 149
\font . . . 1, 2, 3, 141, 632, 633, 1528, 2526, 3129
font name 199, 200
font name 200
fontdef.dtx (class) 262
\fontencoding 1, 2542, 2543, 2544, 2545
\fontfamily 197, 1356, 2561
fontinst (package) 193
fonts
Arial a799
Arial Unicode MS 204
cmr 194
cmss 194
cmtt 194
code2000 203
code2001 203
Dunhill 206
family 194
Fibonacci 206
Karl Berry scheme 204
non-serif 191
pi fonts 203
serif 191
sizing commands 209
Symbola 203
wingdings 203
Zapf Dingbats 202
\fontseries 1356, 2561
fontspec (package) 208
FOO 602
\FOO 36, 37
foo commands:
\foo_underscore:N . . . 1984
foocounter counter 747
\footins 1
footitle 756
\footnote 383
\footnote 224, 381, 423, 2536
\footnotechanges 224
\footnotemark 381, 384
\footnoterule 590
\footnoterule 1
footnotes
\footnoterule 590
kernel \footnoterule 382
minipage (kernel) 382
\footnotesep 1
\footnotesize 27, 47, 208, 213, 224, 239, 262, 268, 685, 773, 1019, 1131, 1166, 1373, 1439, 1706, 1993, 1999, 2010, 2263, 2402, 2460, 2979
\footnotetext 381
\footnotetext 1, 382
\foreword 3
\forewordname 3
format 886

format commands:
\format_block:nnnn 95, 206, 310
\format_display:nnnn . 93, 307
\format_hang:nn 63, 99, 102
\format_hang_inmargin:nn 96, 103, 104
\format_inline:nnn ... 314
\format_inmargin:nnn . 52, 106
\format_part_traditional:nn 107, 321
\format_traditional:nn 109, 346
\format_block:nnnn 897
\format_display:nn 901
\format_hang:nn 895
\format_hang_inmargin 896
\format_inmargin 895
\fox 193
\frac 270
\frame 1
\framebox 309
\frameit *423*
\frenchspacing 19
\frontmatter 573
function environment 807
function (environment) .. 491, *804*
function commands:
\function_one: *805*
\function_two: *805*
\fussy 412
\fvset 7

G

\g@addto@macro 519
\gcd(gcd) a783
\gdef 206, 224, 229
\GenericError 64
\GenericInfo 64
\GenericWarning 64
\GetFileInfo 233, 1718
\getlinechar 1
\getpagerefnumber ... 1526, 1535
\getridofats ... 2685, 2691, 2698
\getridtest 2686, 2687
\global 227, *440*
\glossary *400*
\glossary 1
\glossaryname 2371
\GlossaryParms 2526
\GlossaryPrologue 2523
\Gluon 2165
glyph 193
gobble 1080
\goodbreak 1
\graphene 2166
\graviton 2167
\graybox 770
\graybox ... 1935, 1950, 1951, 1952
group commands:
\group_begin . 102, 169, 189, 323, 334, 348, 368, 805, 1030, 1037, 2056, 2075
\group_end 105, 195, 195, 338, 341, 372, 375, 810, 1033, 1042, 2069, 2163
\groupcolor 1821

H

h, George Douros 204
hacks
escapechar 45
\hang *24*
\hangafter 1533, 3152
hanja 208
\HARM 2316
harmony (package) 779
\HARMtrue 2318
\hb@xt@ 36
\hbond 2168
hbox commands:
\hbox:n 1230
\hbox_gset 1112
\hbox_gset_end 1127
\hbox_set 1304
\hbox_set_end 1309
\hbox_unpack_clear . 1262, 1305
\hboxcolor 1836
hcoffin commands:
\hcoffin_set:Nn .. 948, 963
\headheight 1
\headsep 1
\headtoname 2367
\Hermaphrodite 24, 25
\hfill 239
\hglue 22
\hideoutput 1
\hideskip 1
\hidewidth 1
\Higgsboson 2169
\hline 1009, 1013
\hole 2170
\hom(hom) a783
\hookleftarrow 2225
\hookrightarrow 2226

\hphantom 264
\href 2473, 2493, 2509, 2535
\hrule 2961
\hrulefill 25
\hsize *423*, 2898
\hskip *378*
\hspace 78, 88
\hspace *78*
\hspace* 78
\hspace* *78*
\Huge 1
\huge 1
\hyperlink 642, 645, 649
\hyperref 759
hyperref (package) 196
\hyperref 1334, 1533, 1550
\hypersetup 1147
\hypertarget 516, 524, 532
\hyphenchar . 632, 633, 2526, 3129
hànzì 208

I

\ialign 1
\idotsint($\int\cdots\int$) a781
\idxboth 1749, 1753, 2015
\idxbothbegin 1750, 1754
\idxbothend 1751, 1755
\idxCTAN 2539
\idxlanguage 764
\idxlanguage 1745
\idxTBsyms 2468
\ier 324, 325
\if *499*
if commands:
\if_case:w 180, 223, 267, 284, 336, 379, 459, 506, 549, 593, 639, 682, 726, 768, 812, 854, 902, 918, 934, 1049, 1091, 2233, 2266
\if_dim:w 9, 175
\if@compatibility 507
\if@endpe 295
\if@eqnsw 270
\if@fcolmade 442
\if@filesw 94
\if@firstcolumn 442
\if@inlabel 288
\if@insert 442
\if@mparswitch 443
\if@multiplelabels 249
\if@newlist 289
\if@newlistfalse 289
\if@newlistfalse 289
\if@newlisttrue 289
\if@newlisttrue 289
\if@nmbrlist 289
\if@nobreak 81
\if@noitemarg 289
\if@noitemargfalse 289
\if@noitemargtrue 289
\if@noparitem 289
\if@noparlist 289
\if@noparlistfalse 289
\if@noparlisttrue 289
\if@partsw 94
\if@reversemarginpar 443
\if@specialpage 442
\if@twocolumn 443
\if@twoside 443, 568
\ifARR 2213
\IfBooleanTF 29, 124, 151, 181, 293, 1475, 1521, 1565, 1575, 1585
\ifcase 213
\ifdefined 1626
\ifdim 133, 2940
\ifDOZ 2052
\ifEDICE 2026
\iffancyvrb 3, 4
\ifFC 2355, 2356
IfFileExists 491
\IfFileExists 100
\IfFileExists . 3, 175, 1637, 2028, 2054, 2088, 2215, 2317
\ifHARM 2315
\ifhmode 2987, 3125
\IfNoValueTF . . . 1792, 1813, 1824, 1840, 1858, 1869, 1880, 1890, 1980, 2003, 2274, 2310, 2390, 2445
\ifnum 178
\ifodd 280
\IfRefUndefinedExpandable 1530
\ifshowsymtable 2109, 2110, 2139, 2624, 2625, 2646
\ifSVR 2085
\IfValueTF 41, 68, 1730
\ifVIET 2368, 2369
\ifvmode 56
\ifx 185, 230
\ignorespaces *283*
\ignorespacesafterend . . 1, 501
\iiiint($\iiiint$) a781
\iiint($\iiint$) a781
\iint($\iint$) a781
\immdiate *94*
\immediate 183, 187, 188, 204, 207
implementation (environment) 466

in commands:
in_callback 1
\in_callback *545*
\include 97
\include *93, 94, 98*, 179, 1641
\IncludeInRelease 1
\includeonly 97
\includeonly *93, 94*
\incsyms 770
incsyms counter 772, 779
\incsyms ... 1932, 1957, 1964, 1965, 1971, 1976, 1992, 1998, 2009, 2074, 2082, 2200, 2236, 2244, 2249, 2250, 2251, 2254, 2255, 2257, 2261, 2268, 2281, 2286, 2290, 2291, 2292, 2294, 2298, 2304, 2330, 2344, 2345, 2349, 2351, 2353, 2357, 2360, 2370, 2378, 2384, 2399, 2409, 2413, 2414, 2416, 2421, 2422, 2423, 2424, 2426, 2430, 2434, 2435, 2437, 2440, 2441, 2450, 2451, 2453, 2458
\index 389
\index *400*
\index 1
index doc settings 749, 758
index format 763
\indexaccent 769
\indexaccent .. 1906, 2298, 2304, 2331, 2349, 2353, 2357, 2370, 2426, 2430, 2437
\indexboth 765
\indexboth 1753
\indexbothbegin 1754
\indexbothend 1755
\indexcommand 766
\indexcommand .. 1757, 1761, 1763, 1788, 1906, 1993, 1999, 2074, 2082, 2237, 2245, 2249, 2250, 2251, 2254, 2255, 2262, 2268, 2281, 2286, 2294, 2351, 2379, 2385, 2400, 2409, 2413, 2414, 2417, 2422, 2423, 2434, 2435, 2440, 2441, 2450, 2451, 2453, 2459, 2658, 2664, 2667
\indexcs 716
\indexcypriot 767
\indexcypriot .. 1810, 1964, 1965
\indexDing 781
\indexDing 2397
\indexentry 208, 1690
\indexenv 724
Indexing
customizing 396
doc 390
general 390
multi-page 390
Indexing 390
\indexlinearb 767
\indexlinearb 1837
\indexmany 764
\indexmany 1727
indexmany commands:
\indexmany: 1729, 1732, 1738
\indexmathcmd .. 1888, 3192, 3196, 3199, 3202, 3205, 3209, 3213, 3219, 3224, 3228, 3232, 3238
\indexname 2361
\indexoption 2719
\IndexPrologue 2293
\indexpunct 770
\indexpunct 1771, 1773, 1909, 2010, 2291, 2292, 2421, 2424, 2657
\indexsoutharabian 769
\indexsoutharabian 1876
\indexsoutharabian 1877
\indexstaves 767
\indexstaves 1821, 1971, 1976
\indexTextcomp 782
\indexTextcomp 2415
\indexugar 768
\indexugar 1855
\indexvarpersian 768
\indexvarpersian 1866
\inf(inf) a783
\input 101
\input *55, 93, 493*
\InputIfFileExists 101
\InstanceKey 1586
\InstanceSemantics 1587
int commands:
\int_case 2508
\int_compare ... 55, 111, 341, 343, 345, 349, 354, 778, 1047, 1233, 1337, 1414, 1454, 1473, 1476, 1640
\int_eval 1236
\int_gdecr 1308
\int_gincr 1306, 1932
\int_incr 1213, 1322
\int_new 25, 49, 50, 67, 85, 1931
\int_set ... 1471, 2282, 2285
\int_to_Alph_finnish . 949
\int_to_alph_finnish . 984
\int_to_Alph_slovenian 1844

\int_to_alph_slovenian 1876
\int_to_symbols .. 951, 986, 1846, 1878
\int_use ... 1006, 1691, 1933
\c_zero *871*
\intarraycolor 1842
\intcolor 1820
integrals 781
\integralsymbol 800
\integralsymbol 3203
\interaction 2171
\interdisplaylinepenalty . 263
\interfootlinepenalty 19
\interlinepenalty 86, 133, 138, 325, 350, 2984, 2987, 3122, 3125
\internalsym 2172
\intextsep 481
\ion 2173
\ionicbond 2174
iow commands:
\iow_char 686, 688, 689
\iow_close 2215
\iow_indent 2312, 2533
\iow_log 422
\iow_new 2164
\iow_newline 2173, 2184, 2189, 2198, 2204, 2226, 2227, 2228, 2243, 2249, 2254, 2256, 2267, 2272, 2273, 2275
\iow_now ... 2185, 2190, 2199
\iow_open 2174
\iow_term 2173, 2216
\iow_term:n *864*
\iowcolor 1824
\ishortstack 1
\item 295
\item *24*
\itemcalendaroption 2724
\itemindent 287
\itemindent 1334, 2730
\itemitem *24*
itemize 299
itemize environment 299
\itemsep 283, 287
\itemsep .. 1335, 1344, 2552, 3033, 3097
\iterate 1
\itshape . *197*, 652, 656, 1039, 3036

J

\jot 263
\Jpsimeson 2175

K

\K 772
\k *129*
kanji 208
\Kaonminus 2176
\Kaonnull 2177
\Kaonplus 2178
Karl Berry Scheme 204
\Kcyp 771
\Kcyp 1966
\KED 774
\KED . 2018, 2019, 2020, 2021, 2022, 2023, 2073
\keepsilent 494
\ker(ker) a783
kernel
lists
\item 282
\list 291
kernel-ltmath.dtx (class) 262
\kernel@ifnextchar 52
keymarglist environment 797
Keys
/errors/boolean expected .. 670
/errors/unknown choice value 670
/errors/unknown key 670
/errors/value forbidden ... 669
/errors/value required ... 669
/utils/exec 669
doc left 749
doc right 749
/foo/
footitle 756
/key/
font name 199
/phd/
bibliography font 205
font name 199, 200
font name 200
/phdd/
codecolorize 859
index doc settings 758
keys commands:
\keys_define . 538, 820, 1136
\keys_set 551, 894, 1203
\keyscolor 1833
\keyval 795
\keyval *795*, 3014
\keyvaluestyle 1743
\keywordsregex 1900
keywordsregex 1900

keywordsregex commands:
\keywordsregex_aux:n 1878, 1897
\Kfeyn 2082
\Kgr 2253
\kill 1591
\KN 2254
\KNbig 2255
\Knoidx 2257
KOMA (class) 210
\KOMAoptions 102
\Kp 2007
\Kpig 2200
\Kr( ) a779
Kroll 389
Kroll
Leon Kroll 389
\Ks 2234
\Kstav 771
\Kstav 1979
Kstav commands:
\Kstav_no_opt_arg 1974, 1982
\Kstav_opt_arg . . 1969, 1985
\Kt 2242
\Kv 777
\Kv 2249

L

l internal commands:
\l__phd_make_new_heading_keys:n . 198, 417, 423, 424, 425, 426, 427, 428, 429, 430
\l__phd_make_new_numbering_keys:n 296, 418
\l__phd_set_headings_key_defaults:n . 435, 494, 495, 496, 497, 498, 499, 500, 501, 502
\l@ngrel@x 40
\l_@@_make_new_heading_keys:n 885
\l_@@_make_new_numbering_keys:n 888
\l_@@_set_headings_key_defaults:n 891
\l_phd_months_abbreviated_ 10
\l_phd_months_narrow_ 10
\l_phd_months_wide_ 9
\label 249
\label 52, *247*, 865, 991, 1065, 1079, 1385, 1409, 1433, 1440, 2016, 2152, 2222, 2348
label commands:
\label_number_box 142, 239
\labelargcount 2542, 2547
\labelargcount 2541
\labelenumiv *298*
\labelitemi 298
\labelitemi 1
\labelitemii 298
\labelitemii 1
\labelitemiii 298
\labelitemiii 1
\labelitemiv 298
\labelitemiv 1
\labelsep 287
\labelsep 2544, 3079, 3089
\labelwidth 287
\labelwidth 3078, 3088
\language 292, 634
\LARGE 1, 274
\Large 1
\large 1, 457
\lastbox 187, 226
\LastDeclaredEncoding 162
\lastskip *84*
\LaTeX . 1, *90*, *92*, *104*, *145*, *152*, *157*, *176*
LaTeX kernel
spacing
\newline 80
LaTeX kernel classesFile n
ltlength.dtx 153
Latex kernel@normalsize 487
Latex kernel@setfontsize 487
\LaTeXe 1, *90*, *92*, *104*, *145*, *152*, *157*, *176*, 2355
\latexe 2575, 2582, 2590
\latexreleaseversion 1
\lbrack 20
\ldots *5*, 352, 361, 370, 378
\leavevmode 24
\leftarrow 2227
\lefteqn 271
\leftline 315
\leftmargin 287
\leftmargin . . . 1333, 2549, 2730, 3080, 3090
\leftmargini 288
\leftmarginii 288
\leftmarginiii 288
\leftmarginiv 288
\leftmark 1
\leftrightarrow 2228
\leftskip *284*
\Lefttorque . . 244, 273, 367, 1017
\len 3060
\let 186, 189, 190, 191, 192, 193, 194, 196, 199, 200,

201, 202, 205, 210, 221, 226, 227, 241, *454*, *499*
\levelchar 1522, 2386, 2433, 2486, 2813, 2815
\lg(lg) a783
\lim(lim) a783
\liminf(lim inf) a783
limits 783
\limsup(lim sup) a783
\line 337
Linear B 768
Linear B
\BPamphora (𐄁) a768
\BParrow (𐄐) a768
\BPbarley (𐄢) a768
\BPbilly (𐄣) a768
\BPboar (𐄤) a768
\BPbronze (𐄥) a768
\BPbull (𐄦) a768
\BPcauldroni (𐄧) a768
\BPcauldronii (𐄨) a768
\BPchariot (𐄩) a768
\BPchassis (𐄪) a768
\BPcloth (𐄫) a768
\BPcow (𐄬) a768
\BPcup (𐄭) a768
\BPewe (𐄮) a768
\BPfoal (𐄯) a768
\BPgoat (𐄰) a768
\BPgoblet (𐄱) a768
\BPgold (𐄲) a768
\BPhorse (𐄳) a768
\BPman (𐄴) a768
\BPnanny (𐄵) a768
\BPolive (𐄶) a768
\BPox (𐄷) a768
\BPpig (𐄸) a768
\BPram (𐄹) a768
\BPsheep (𐄺) a768
\BPsow (𐄻) a768
\BPspear (𐄼) a768
\BPsword (𐄽) a768
\BPwheat (𐄾) a768
\BPwheel (𐄿) a768
\BPwine (𐅀) a768
\BPwineiih (𐅁) a768
\BPwineiiih (𐅂) a768
\BPwineivh (𐅃) a768
\BPwoman (𐅄) a768
\BPwool (𐅅) a768
\linebreak 79
\linethickness 335
\linethickness 1, *329*
\linewidth 287
\linewidth . 259, 288, 2627, 2898, 2944, 2948, 2949, 2955, 2969, 3008
\list 291
\list 2547, 2729, 3076, 3086
\listfigurename 591
\listfigurename 2359
\listfiles 103, 486
\listfiles *55*
\listi 596
listings commands:
\listings_code . 2014, 2094, 2108, 2130
Listings Styles
extended *731*
highlite *732*
simple *729*
simplex *730*
singleline *730*
\listparindent 287
\listtablename 2360
\llap 1
\ln(ln) a783
\LoadClass 514
\LoadClass 501
\LoadClass 88, 93, 97, 101, 106, 109, 115, 125
\LoadClassWithOptions 515
\LoadClassWithOptions *505*
\log(log) a783
log-like symbols 783
\longhookrightarrow 2229
\longleftarrow 2230
\longleftrightarrow 2224
\longmapsfrom 2225
\longmapsto 2226
\longrightarrow 2227
\loop 1
\lorem 11, 13, 15, 274
\loremlines 1
\Lpack 2497, 2513
\lphdparagraphnumberprefixtl 64
\lphdparagraphnumbersuffixtl 65
\lphdsubparagraphnumberprefixtl 66

\lphdsubparagraphnumbersuffixtl 67
\lphdsubsectionnumberprefixtl 60
\lphdsubsectionnumbersuffixtl 61
\lphdsubsubsectionnumberprefixtl 62
\lphdsubsubsectionnumbersuffixtl 63
\lq 1
lrbox 307
\lstdefinelanguage 190, 417, 471, 746
\lstdefinestyle .. 225, 256, 308, 344, 659, 1010, 1154
\lstloadlanguages 197
\lstnewenvironment .. 298, 303, 324, 379, 385, 731, 775
\lstset 202, 299, 304, 325, 329, 331, 335, 380, 381, 386, 387, 732, 776, 777
\ltindex 2147, 2148, 2149, 2150, 2151
ltxdoc (class) 390, 726, 746
\ltxsyntaxfont 3044
\ltxsyntaxlabelfont 3046
\ltxtoday 5
\luacmd 3135
\luaexample 962
\luafunction *541*
\luatex 2056

M

\m@ne 11
macro environment 807
macro (environment) 491, *805*
\macroarguments 2546
\MacroFont 1338
\MacroLongFont 1338, 1354
\MacroTopsep 1256
\magicequal 2549
\magicequalname 2550
\magicrbrack 2548
\magicVertname 2360, 2552
\magicvertname . 1914, 1920, 2551
\magnon 2179
\magstep 1
\magstephalf 1
Mahjong 204
main
xindy 398
\main 1703
\mainmatter 573
make commands:
\make_new_bool:n . 259, 271, 272, 273
\make_new_opening_keys:n 276, 288, 289, 290
\makeatletter 54
\makeatother 54
\makebox 304
\makebox 105, 106, 653, 2564, 3190, 3195, 3198, 3201, 3204, 3207, 3208, 3211, 3212, 3216, 3217, 3222, 3223, 3227, 3231, 3235, 3236
\makeglossary *400*
\makeglossary 1
\makeindex 389, 400
\makeindex *400*
\makeindex *400*
\makelabel 293, 298
\makelabel *298*
\MakePercentComment 1620
\MakePercentIgnore 1618
\MakePrivateLetters 723
\MakePrivateLetters 7, 395, 1491, 2290
\MakeShortVerb 139
\maketitle *357*
\MakeUppercase 150
\maltese(✠) a782
\manual 1528, 1529
\mapsfrom 2228
\mapsto 2229
\marg 747
\marg 159, 447, *804*, *822*, 1035, 2203
\marginpar 379
\marginpar *378*
\marginparpush *378*
\marginparsep . *378*, 974, 978, 995
\marginparwidth .. 149, *378*, *682*, 952, 978, 1110
\marglist 3096
\marglistfont 3036, 3081
\marglistsep .. 3064, 3071, 3079, 3089
\marglistwidth 3066, 3069, 3078, 3088
\margnotefont 3040
\margoptionfont 3038, 3091
\mark 1, *410*
\markboth 1
\markright 1
\marks 529
math (environment) 1

math commands:
\c_math_operator_regex 1950, 1963
\mathdollar 1
\mathellipsis 1
\mathhexbox 1
\mathindent 272
\mathoperatorcolor 1952
\mathpalette 264
\mathparagraph 230
\mathparagraph 1
\mathsection 230
\mathsection 1
\mathsterling 1
\mathstrut 265
\mathsurround *91*
\mathunderscore 1
\mathversion 166
\mathversion *166*
\matrix 265
\max(max) a783
\maxdimen 1
\maxwellDistrib 2180
\maya 2413
\mayadigit 2413
\mbox 304
\medbreak 1
\medmuskip 231
\medskip 86
\medskip .. 2633, 2674, 2916, 2995
\medskipamount 87
\medskipamount 320, 321, 372, 373, 411, 412, 742, 743, 978, 991, 1164, 1165
memoir (class) 210
\MessageBreak 64
\meta 746
\meta . 160, 162, 164, 432, 445, 448, 450, 451, *804*, *804*, 940, 1026, 1029, 1046, 1410, 3175
\metalbond 2181
\method 2182
\midrule 1088, 2351
\min(min) a783
\minimumgroupdigits 41
\minipage 1
mode commands:
\mode_if_math:TF . 565, 628, 938
\mode_leave_vertical: 1131
\Module 2557
\module_error *544*
\module_info *544*
\module_warning *544*
monetary symbols 765, 788
\MoonPha *765*
\month . 144, 172, 218, 262, 328, 374, 417, 443, 501, 544, 586, 633, 677, 720, 764, 807, 850, 893, 1044, 1087, 1128, 1155, 1183, 1213, 1244, 1272, 1311, 1341, 1375, 1418, 1460, 1502, 1531, 1575, 1605, 1634, 1663, 1690, 1727, 1756, 1783, 1810, 1838, 1929, 1956, 1984, 2023, 2060, 2088, 2116, 2156, 2198, 2266, 2284, 2339
\MoonPha 2414
msg commands:
\msg_error 723, 872, 882, 919, 1105
\msg_info 79, 2316, 2538
\msg_new 73, 362, 370, 373, 376, 381, 386, 389, 2309, 2530
\msg_warning 188, 1141
\msgcolor 1827
\multicolumn 1091, 1097, 2154, 2158
\multiput 330
\multiput 1, 1
\muon 2183
musical symbols 779
\myempty 2682, 2687, 2696
\mygobble 2683, 2701
\myprintocmmand 2714

N

\N 778
\name *36*
names commands:
\z_make_names 112, 127
\narrower 24
\nathdouble 2281, 2282
\nathtriple 2287
\Nbig 2294
\nbitem 298
\nearrow 2230
\NeedsFormat *515*
\Needspace 2112
\NeedsTeXFormat 515
\NeedsTeXFormat 1, 2, 5, *111*
\negthinspace 78, 89
\neutrino 2184
\neutron 2185
\new@command 40
\new@environment 44
\new_attribute *542*
\new_bytecode *542*
\new_chunkname *542*

\new_luafunction *542*
\new_whatsit *542*
\newattribute *541*
\newboolean 486
\newbox 1, 2907
\newcatcodetable *541*
\newcommand 40
\newcommand 157, 171
\newcount 12, 148
\newcounter 147
\newcounter 782, 785, 2541
\newdimen 1
\NewDocumentCommand 267, 458, 460, 462, 464, 512, 514, 522, 530, 1198, 1473, 1501, 1736, 1788, 1802, 1810, 1821, 1837, 1855, 1866, 1877, 1888, 1979, 2001, 2272, 2308, 2388, 2441, 2443, 2465, 2532
\NewDocumentEnvironment . 466, 473, 497, 503, 3241
\newenvironment 44
\newfontfamily 141
\newhelp 1
\newif 45
\newif ... 3, 7, 12, 1049, 1050, 1051, 2026, 2052, 2085, 2202, 2213, 2222, 2315, 2355, 2368, 2906
\newinsert 16
\newlabel 249
\newlanguage 1
\newlength 153
\newlength 1940, 1941, 1942, 3013, 3064, 3066, 3074
\newline 80
\newluabytecode *541*
\newluachunkname *541*
\newluafunction *541*
\newmarks 1
\newmuskip 1
\newpage 435, 444
\newpage *433, 434*
\newread 1, *37*
\newrobustcmd 128, 1211, 1216, 1227, 1232, 1236, 1241, 1251, 1833, 3058, 3060, 3062
\newsavebox 306
\newsavebox 2555
\newskip 153
\newskip 1
\newtcolorbox 198, 200, 1360
\newtheorem 1
\newwhatsit *541*
\newwrite 1
\nfss@catcodes 169
\nkonumber 1417, 1432
\Nn 778
\Nn 2280
\Nnt 2285
\nobreak 22
\nobreak *365, 384*
\nobreakdashes 87
\nobreakspace 87
\nobreakspace 330, 357, 1341
\nobreakspace␣ *88*
\nocite 405
\nocite *403*
\node 361, 362
\noexpand *382*
\nofiles 96
\noindent . 211, 352, 361, 370, 378, 909, 955, 957, 958, 996, 1438, 1532, 1621, 2626, 2715, 2716, 2943, 2959, 3189, 3194, 3197, 3200, 3203, 3206, 3210, 3215, 3221, 3226, 3230, 3234
\noindexing 1614, 2762
\nointerlineskip 1, 251
\nolinebreak 79
\nolinkurl 452
\nonfrenchspacing 19
\nonumber 271
\nopagebreak 79
\nopagebreak *76*
\noprotrusion 1
\normalbaselines 20
\normalbaselineskip 1
\normalcolor 1, *479*, 1911
\normalfont 214
\normalfont 316, 457, 488, 515, 1015, 1943, 1944, 1945, 2979, 3048, 3134
\normallineskip 1
\normallineskiplimit 1
\normalmarginpar 1, *378*
\normalsfcodes 96
\normalsfcodes *455*
\normalsize 243, 487
\normalsize 488, 515, 1015
\notpredefinedmessage ... 2580
\notpredefinedmessageABX 2588
\Np 2293
\nucleus 2186
\null 21
\number 154, 209
\numberline . 1, 73, 78, 2124, 2632
\numberlineat 731
\numberlineat 332, 333

\numexpr 789, 798, 878
\nwarrow 2224

O

\o@lign 25
\oalign 25
\oarg 747
\oarg 161, 447, *804*, 1044
\obeycr 89
\obeylines 21
\obeyspaces 21
\obeyspaces 1119
\occitanday 1545, 1549
\occitanmonth 1531, 1549
octavo (class) 485
\oddsidemargin 150, 3069
\of 270
\offinterlineskip 1
\Ohne(') a779
\oldherm 24
\on@line 70
\onecolumn 444
\OnlyDescription 1594
\ooalign 25
\open *423*
openbib 601
\openout 187, 1659
\openup 266
\operator@font 230
\operator@font 262
\opt 2499
\option 797
\option 3104
\optionlistfont 3042
\OptionNotUsed 512
\optname 2504
\or 213, 214, 215, 216, 217, 218
or commands:
\or: 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 727, 728, 729, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278
\orbit 2187
\origDeclareMathSymbol . 2091, 2101
\origDeclareSymbolFont . 2090, 2100
\origepsdice 2031, 2032
\orignewcommand 2319, 2324
\origProvidesPackage ... 2045, 2046, 2048
\origSetSymbolFont . 2092, 2102
\othernumberingsystems 39
\outerparskip 287
\output 447
\output *423*
\oval 1
\overfullrule 412
\ovidius 359

P

\p@ 18
\PackageError 67
\PackageError 2861
\PackageInfo 67

Packages
anyfont 210
moresize 210
xkeyval 642
packages
background 427
bobhook 424, 425
calc 155
draftwatermark 425
everypage 425, 427
everyshi 424, 425
flippdf 425
fontinst 193
fontspec 208
harmony 779
hyperref 196
phd 191
phd-pkgmanager 427
picture 328
prelim2e 426
sectsty 910
textcomp 765
uncial 201
watermark 425
xkeyval 486
Packages:
amsthm *355*
babel *245*
biblatex *403*
bobhook *425*
datetime2 . . . . . *1*
Doc *498*
doc *390*, *499*, *675*, *721*, *724*, *843*, *850*, *851*
doc.sty *826*
DocStrip *493*
docstrip *390*, *493*
environ *724*
etoolbox *60*, *724*
everypage *424*
expl3 . *60*, *665*, *804*, *806*, *806*, *850*, *870*, *871*
fancyvrb *821*
fixltx2e *438*
fontspec *191*
fonttable *207*
hypdoc *784*, *867*
hyperref . *247*, *644*, *759*, *828*, *844*
imakeidx *397*, *398*
keyval *642*
kvoptions *644*
l3basics *817*
l3doc *805*, *806*, *807*, *853*
l3docstrip *803*, *803*
l3keys *870*
Lazarides *784*
listings *721*, *728*, *741*
ltxdoc *675*, *803*, *804*, *804*, *804*
makeidx *389*, *745*
makeindex *396*
minted *721*
multind *397*
natbib *403*
perpage *245*
pgf *155*, *675*, *724*
pgfkeys *642*
phd *6*, *675*, *721*
phd-colorpalette . *721*, *728*, *746*, *797*
phd-documentation *682*
phddoc *676*
pict2e *333*
picture *332*, *332*
pspicture *333*
refcount *724*, *759*
tcolorbox *303*, *724*, *724*, *724*, *728*, *741*, *886*, *897*
titlesec *489*
titletoc *489*
tufte-book *501*
verbatim *724*
xkeyval *642*, *642*
xkview *642*
xparse *821*, *841*, *904*
xpicture *333*
\PackageWarning 68
\PackageWarningNoLine 67
\pagebreak 79
\pagebreak *434*
\pagegoal *423*
\PageIndex 1639
\pagename 2368
\pagenumbering 245
\pagenumbering *245*
\pageref 247
\pageref *248*, 1481, 1541
\pagestyle 409
palette 886
\panunicode 17, 425, 427, 443, 1373, 1415, 1747, 2021, 2346
\paperheight 442
\paperwidth 442
\par *284*, *285*, *294*, *411*, *416*
\par@deathcycles 292
\paragraph 916
\paragraph 16, 485, 492

paragraph commands:
\paragraph_cntformat . 259
\paragraphmark 1
\parbox 24, 44, 224, *369*, 1704
\parfillskip *284*
\parg 163, 447, *804*
\parindent *916*, *916*
\parsep 287
\parsep . . . 1336, 2550, 3076, 3077, 3082, 3086, 3087, 3092
\parshape *277*, 2949
\parskip 274
\parskip *284*, *416*
\part 574, 911
\part 195, 285, *683*, 1667, 2295, 2525
\partname 167, 2364
\partopsep 283
\PassOptionsToClass 511
\PassOptionsToClass 501
\PassOptionsToClass *505*
\PassOptionsToClass 65, 67
\PassOptionsToPackage 511
\PassOptionsToPackage *505*
\PassOptionsToPackage 59
\path 122
\pattern . . 2819, 2821, 2822, 2823, 2837, 2839
\patternindex 2805
\patternindexinherentlycolored 2826
Patterns
bricks *791*
\pdfpagewidth 212
\pdfstringdefDisableCommands 418, 440
\pdfstringnewline 415
penalties
\interfootline 19
\penalty *434*
\person 2532
pgf 763
\pgfkeys 653
\pgfkeysalso 653
\pgfkeysalso 866, 992
\pgfkeysdeargs 655
\pgfkeysdef 655
\pgfkeysedef 655
\pgfkeysedefargs 655
\pgfkeysgetvalue 652
\pgfkeysifdefined 652
\pgfkeyslet 652
\pgfkeyssetvalue 648, 651
\pgfkeysvalueof 652
\pgfkeysvalueof 147, 149
\pgfmanualdeclare . . 2707, 2708
\pgfqkeys 653
\pgfqkeys 3001
\pgfqkeysalso 653
\phantom 264
\phantom 656
\phantomsection 1372, 1626, 2115, 2630
phd (package) 191
phd commands:
\phd_after_heading 91, 137, 178, 212, 216
\phd_brackets 1323, 1407, 1412
\phd_doc_do_description 1350, 1386, 1413, 1442
\phd_float_box 147, 196, 242, 245, 253
\phd_format_router . 85, 89
\phd_hang_from . . . 6, 65, 137
\phd_index_col . . . 141, 1302
\phd_index_env . 1271, 1408, 1466, 1636
\phd_index_key . 1431, 1486, 1505
\phd_index_key:n 1480
\phd_index_key_path 1288, 1436, 1508
\phd_index_print_c . . 130, 1189, 1200, 1218, 1232, 1243
\phd_index_val 157, 176
phd_index_val 156
\phd_manual_top 1342
\l_phd_months_abbreviated_finnish 901
\l_phd_months_narrow_finnish 917
\l_phd_months_wide_abbreviated 283
\l_phd_months_wide_albanian 378, 417
\l_phd_months_wide_basque 505, 544
\l_phd_months_wide_brazil 547
\l_phd_months_wide_breton 591, 633
\l_phd_months_wide_bulgarian 637, 677
\l_phd_months_wide_catalan 680, 720
\l_phd_months_wide_croatian 724, 764
\l_phd_months_wide_czech 767, 807

\l_phd_months_wide_danish 811, 850
\l_phd_months_wide_estonian 853, 893
\l_phd_months_wide_finnish 933, 1044
\l_phd_months_wide_french 266, 328
\l_phd_months_wide_friulian 1048, 1087
\l_phd_months_wide_galician 1090
\l_phd_months_wide_german 222, 262
\l_phd_months_wide_greek 179, 218
\l_phd_months_wide_italian 335, 374
\l_phd_months_wide_russian 458, 501
\phd_more_keywords_tl 1787, 1797, 1798
\phd_print_col 133, 140, 146
\phd_print_com . 1205, 1382
\phd_print_env . 1222, 1407, 1412, 1465, 1471
\phd_print_key . 1430, 1435, 1481, 1504, 1507, 1514, 1516
\phd_print_key:n 1480
\phd_ref_doc 1528, 1556, 1570, 1581
\phd_ref_doc_star .. 1548, 1561, 1571, 1582
\phd_set_opening . 309, 348
\phd_setparent 405, 408, 409, 410, 411, 412, 413, 414

phd internal commands:

__phd_book_divisions_clist .. 114, 179, 415, 420, 492
\l__phd_chapter_afterskip_tl 394
\l__phd_chapter_align_tl 303, 402
\l__phd_chapter_beforeskip_tl 392
\l__phd_chapter_color_tl 403
\l__phd_chapter_fontfamily_tl 300, 399
\l__phd_chapter_fontshape_tl 302, 401
\l__phd_chapter_fontsize_tl 301, 400
\l__phd_chapter_fontweight_tl 299, 398
\l__phd_chapter_indent_tl 390
\l__phd_chapter_label_fontfamily_tl 166
\l__phd_chapter_label_fontshape_tl 167
__phd_create_new_element:nn 162, 181, 188
__phd_document_type_tl 69, 72
__phd_document_types_clist 68, 74
\l__phd_paragraph_afterskip_tl 497
\l__phd_paragraph_align_tl 504
\l__phd_paragraph_beforeskip_tl 496
\l__phd_paragraph_color_tl 505
\l__phd_paragraph_fontfamily_tl 501
\l__phd_paragraph_fontshape_tl 503
\l__phd_paragraph_fontsize_tl 502
\l__phd_paragraph_fontweight_tl 500
\l__phd_paragraph_indent_tl 495
\l__phd_paragraph_number_after_tl 253, 259
\l__phd_part_afterskip_tl 295
\l__phd_part_beforeskip_tl 293
\l__phd_part_color_tl 304
\l__phd_part_indent_tl 291
\g__phd_saved_heading_tl 83, 194
__phd_secondary_clist 129, 130, 186
__phd_secondary_clist_names_clist 119, 120, 125
\l__phd_section_afterskip_tl 419
\l__phd_section_align_tl 427
\l__phd_section_beforeskip_tl 417
\l__phd_section_color_tl 428

\l__phd_section_fontfamily_tl 424
\l__phd_section_fontshape_tl 426
\l__phd_section_fontsize_tl 425
\l__phd_section_fontweight_tl 423
\l__phd_section_indent_tl 415
__phd_set_document_divisions .. 78, 82, 88, 94, 100, 106
__phd_set_document_type:n 70, 77
\l__phd_subparagraph_afterskip_tl 523
\l__phd_subparagraph_align_tl 527
\l__phd_subparagraph_beforeskip_tl 522
\l__phd_subparagraph_color_tl 528
\l__phd_subparagraph_indent_tl 521
\l__phd_subsection_afterskip_tl 444
\l__phd_subsection_align_tl 448
\l__phd_subsection_beforeskip_tl 443
\l__phd_subsection_color_tl 449
\l__phd_subsection_indent_tl 442
\l__phd_subsubsection_afterskip_tl 469
\l__phd_subsubsection_align_tl 477
\l__phd_subsubsection_beforeskip_tl 468
\l__phd_subsubsection_color_tl 478
\l__phd_subsubsection_fontfamily_tl 473
\l__phd_subsubsection_fontshape_tl 475
\l__phd_subsubsection_fontsize_tl 474
\l__phd_subsubsection_fontweight_tl 472
\l__phd_subsubsection_indent_tl 467
phd-pkgmanager (package) 427
phd@doc@head environment ... 755
\phd_after_heading: 909
\phd_book_divisions_clist 882
\phd_create_new_element:nn 884
\phd_document_type_tl 881
\phd_float_box:nnn 900
\phd_format_router:nnn ... 906
\phd_index_col 753
\phd_index_env 753
\phd_index_key 753
\phd_print_com 751
\phd_ref_doc 759
\phd_secondary_clist 882
phdcl internal commands:
\g__phdcl_article_bool 12, 94
\g__phdcl_book_bool 11, 89
\g__phdcl_checkfunc_bool 41, 43
\g__phdcl_checktest_bool 45, 47
\g__phdcl_cs_break_bool 57
\g__phdcl_doctype_tl . 60, 62, 64, 85, 125
\g__phdcl_kernel_bool 49, 51
\g__phdcl_ldoc_bool ... 18
\g__phdcl_ldociii_bool 19
\g__phdcl_lmodern_bool 53, 55
\g__phdcl_macrocode_colorize_bool 10, 22
\g__phdcl_masterthesis_bool 17
\g__phdcl_report_bool 13, 98
\g__phdcl_scrartcl_bool 15, 107
\g__phdcl_scrbook_bool 14, 103
\g__phdcl_scrreprt_bool 16
\phdcolor 1838
\phdcs 747
\phdcs 1023, 1200, 1207, 1384, 1406, 1411, 1592
phdd commands:
\phdd_code_colorize_bool . 23, 24, 39, 1979, 2076, 2086, 2092, 2106, 2127, 2128
\phdd_doc_color_aux:n 127
\phdd_doc_color_star:n 127
\phdd_index_key:n ... 1480
\phdd_index_typeset_bool 97, 110

\phdd_key_color_color_tl 68, 130, 135, 1253
\phdd_key_color_command_tl . 64, 103, 1007, 1200, 1207
\phdd_key_color_environment_tl ... 65, 1213, 1218, 1224
\phdd_key_color_tl 66, 1229, 1232, 1483
\phdd_print_col 127
\phdd_typeset_bool 99
\phdd_typeset_command 100, 118, 1531
\g_phdd_typeset_documentation_bool 30, 35
\g_phdd_typeset_implementation_bool 31, 36
\phdd_typeset_index_command 108, 123

phdd internal commands:

__phdd_base_form_aux:nnN 268, 292, 318, 717
__phdd_base_form_aux:nnnnnN 306, 308
__phdd_base_form_signature_do:nnn 303
\g__phdd_base_name_tl 78, 747, 748, 754, 760, *812*
\g__phdd_checkfunc_bool 42, *809*, 2171
\g__phdd_checktest_bool 43, *809*, 1421, 2236
__phdd_cmd:nn 421, 424, 428, 545, *821*
\l__phdd_cmd_index_tl 69, 540, 549, 586, 589, *812*
\l__phdd_cmd_module_tl 70, 541, 550, 593, 596, *812*
\l__phdd_cmd_noindex_bool . 71, 542, 547, 584, *812*, *825*
\l__phdd_cmd_replace_bool ... 72, 543, 548, 553, *812*
\l__phdd_cmd_tl 68, 552, 556, 557, 558, 568, 570, 577, 580, 581, 588, 592, *812*
\g__phdd_cs_break_bool 45, 46, 571, *809*
\l__phdd_date_added_tl 82, *813*, 853, 1086, 1089, 1092
__phdd_date_compare:nNnTF 325, 879
__phdd_date_compare_aux:nnnNnnn 325
__phdd_date_compare_aux:w 325
__phdd_date_compare_p:nNn 325
__phdd_date_set:Nn . 853, 854, 866, 878
\l__phdd_date_updated_tl 83, *813*, 854, 1087, 1095, 1098
__phdd_deprecated_on:n 855, 876, 1194
\l__phdd_descr_coffin 19, *807*, 920, 956, 977, 980, 990
\g__phdd_design_colors_tl 1810, 1811, 1846
__phdd_detect_internals:N .. 164, 690, 2088, 2102, 2124, 2144, 2154
__phdd_detect_internals_aux:N 164
\l__phdd_detect_internals_bool 13, 14, 166, *807*
\l__phdd_detect_internals_cs_tl 16, 184, 189
\l__phdd_detect_internals_tl . 15, 173, 174, 175, 176, 177, 179, 181, 182, 184, 185, 186, 187, 190, *807*
\l__phdd_doc_def_tl . 2165, 2176, 2183, 2207
\l__phdd_doc_undef_tl 2166, 2177, 2188, 2210
__phdd_ensuremath_sb:n 607, *826*
\g__phdd_finale_tl . 1605, 1613, 1614
\g__phdd_func_iow .. 2164, 2174, 2185, 2190, 2199, 2215
__phdd_function:nnw 484, 492, *835*, 889
__phdd_function_assemble: 904, 961
__phdd_function_descr_start:w 899, 954
__phdd_function_descr_stop: 903, 954
__phdd_function_end: 489, 493, *835*, 889
__phdd_function_extra_labels: 1010, 1057
__phdd_function_index:n 1017
__phdd_function_init: 892, 916
__phdd_function_label:nN 1021, 1069, 1082

\l__phdd_function_label_clist .. 80, *812*, 859, 932, 1061
__phdd_function_reset: 898, 941
__phdd_function_typeset: 897, 945
__phdd_function_typeset_start: 891, 907
__phdd_function_typeset_stop: 905, 907
\l__phdd_functions_coffin .. 18, *807*, *808*, *834*, *835*, *836*, 923, 948, 950, 973, 994
__phdd_functions_typeout:nN .. 2202, 2208, 2211, 2220
__phdd_get_function_name:n ... 236, 240, 1045, 1278
__phdd_get_function_signature:n 238, 240
__phdd_get_hyper_target:nN . 668, 1063, 1073, 1327, 1478
__phdd_gprop_name:n . 360
__phdd_if_almost_str:nTF .. 109, 117, 264, 568, 2400
__phdd_if_detect_internals_ok:N 196
__phdd_if_detect_internals_ok:NTF 164
__phdd_if_macro_internal:nTF 783, 1302, 1348
__phdd_if_macro_internal_aux:w 783
__phdd_if_macro_internal_p:n 783, 1427, 1467
\l__phdd_in_function_bool 22, *808*, 931, 1128
\l__phdd_in_implementation_bool 73, 469, 476, 482, 487, *812*, *823*
\l__phdd_index_escaped_key_tl .. 2343, 2371, 2372, 2387
\l__phdd_index_escaped_macro_tl 2342, 2373, 2391
\l__phdd_index_internal_bool 59, 603, *811*, 2339, 2446, 2447
\l__phdd_index_key_tl 57, 599, *811*, *869*, *870*, 1310, 2335, 2442, 2445, 2450, 2452, 2455, 2458, 2465, 2466, 2477, 2482, 2485, 2492, 2494, 2505, 2519
\l__phdd_index_macro_tl . 56, 600, *811*, *869*, 2336, 2441, 2443
\l__phdd_index_module_tl .. 58, 595, 601, *811*, *869*, *870*, 2333, 2337, 2444, 2459, 2472, 2473, 2485, 2496, 2497, 2502, 2506, 2519
__phdd_input:n 1615, 1628, 1632
\c__phdd_iow_mid_rule_tl 63
\c__phdd_iow_midrule_tl *811*, 2226, 2228
\c__phdd_iow_rule_tl 61, *811*
\c__phdd_iow_separator_tl 2173, 2216
\g__phdd_kernel_bool . 44, 204, *809*
__phdd_key_drop_underscores: 2468, 2484, 2501
__phdd_key_func: *869*, *870*, 2453, 2480
__phdd_key_get:n 592, 2331, 2439
__phdd_key_get_base:nN 262, *869*, 2441
__phdd_key_get_base_TF:nN 266, 272
__phdd_key_get_module: 2489
__phdd_key_pop: 2451, 2463, 2478, 2483, 2499, 2500
__phdd_key_trim_module:n *814*, 2468, 2486, 2487, 2520
__phdd_key_var: . *869*, *871*, 2456, 2489
\g__phdd_lmodern_bool 41, *809*
\l__phdd_long_name_bool . 23, *808*, *835*, 951, 965, 1109
__phdd_lseq_name:n . 360, 728, *810*, *829*
__phdd_macro:nnw 483, 495, 1199
\l__phdd_macro_argument_tl .. 84, *813*, 883, 893, 1202
\l__phdd_macro_box 65, *811*, *844*, 1228, 1265, 1315, 1317
\l__phdd_macro_documented_tl 40, *809*, 1197, 1225, 1245, 1249
__phdd_macro_dump: 1208, 1254
__phdd_macro_end: .. 488, 496, 1410
__phdd_macro_end_check_tested: 1413, 1417
__phdd_macro_end_style:n 1415, 1436

__phdd_macro_end_wrap_item:n 1441
\l__phdd_macro_EXP_bool .. 35, *809*, 830, 836, 844, 927, 1039, 1167, 1179, 1185, 1219
__phdd_macro_index:nN 1293, 1297
\l__phdd_macro_index_box 66, *811*, 1227, 1262, 1304, 1305
__phdd_macro_init: . 1201, 1211
\l__phdd_macro_int 67, *811*, 1233, 1236, 1322
\l__phdd_macro_internal_bool .. 31, 787, *809*, *831*, 886, 1147, 1153, 1214
\l__phdd_macro_internal_set_bool . 30, 785, *809*, 887, 1148, 1154, 1215
\l__phdd_macro_noTF_bool .. 34, 771, *809*, 850, 926, 1174, 1218
\l__phdd_macro_pTF_bool . . 33, 765, *809*, 842, 925, 1166, 1217, 1431
__phdd_macro_reset: 1209, 1239
\l__phdd_macro_rEXP_bool .. 36, *809*, 831, 837, 845, 928, 1040, 1168, 1180, 1186, 1220
__phdd_macro_save_names: 1206, 1243
__phdd_macro_save_names_aux:n 1246, 1252
__phdd_macro_single:nNN . *827*, *841*, 1274, 1285, 1288
\l__phdd_macro_tested_bool 26, *808*, 1190, 1222, 1360, 1423
\l__phdd_macro_TF_bool 32, 722, 773, *809*, 816, 825, 843, 851, 924, 1161, 1165, 1173, 1216, 1432
__phdd_macro_typeset_block:nN 780, 1272
__phdd_macro_typeset_one:nN 1291, 1313
__phdd_macro_typeset_variant_list:nN 1277, 1282, 1287
\l__phdd_macro_var_bool . 37, *809*, 1158, 1221, 1422, 1474
__phdd_macroname_prefix:n 1342, 1346, 1351
__phdd_macroname_suffix:N 1343, 1352
__phdd_meta:n ... 433, 607
__phdd_meta_original:n 607
\g__phdd_missing_tests_prop 27, *808*, *846*, 1398, 2239, 2258
\g__phdd_module_name_tl . 60, 136, 139, 191, 198, 201, *811*, 1617, 2148
__phdd_names_block_base_map:N 801, 1246
\l__phdd_names_block_tl 51, 698, 731, 733, 741, 803, *810*, *829*, *830*, *831*, *835*, *842*
__phdd_names_get_seq:nN 676, 895, 1204
__phdd_names_parse: 696, 896, 1205
__phdd_names_parse_aux:Nnn 728, 729
__phdd_names_parse_aux:nnn 714, 716, 724, 727
__phdd_names_parse_one:n 696
__phdd_names_parse_one_aux:nnNn 706, 709
\l__phdd_names_seq 54, 700, *810*, 895, 1204, 1399, 1426
__phdd_names_typeset: 739, 1011, 1207
__phdd_names_typeset_auxi:n 739, *830*
__phdd_names_typeset_auxii:n 754, 759, 763, *830*
__phdd_names_typeset_block:nN 767, 772, 773, 776
\l__phdd_names_verb_bool . 53, 679, *810*, 863, 930, 1195, 1223
\l__phdd_nested_macro_int .. 25, 778, *808*, *831*, *841*, 1213, 1414
\g__phdd_nested_names_seq 55, *810*, 1248, 1253, 1451, 1465, 1466, 1469, 1472, 1479, 1484
\l__phdd_no_label_bool 81, *812*, 860, 929, 1059, 1071
\g__phdd_not_tested_seq . . 28, *808*, 1428, 2261, 2277
\l__phdd_output_coffin 17, *807*, 968, 972, 976, 979, 985, 989, 993, 996
\l__phdd_override_module_tl ... 38, 39, *809*, 864, 933, 943, 1196, 1224, 1241, 2332, 2333

__phdd_pdfstring_cmd:w 434
__phdd_pdfstring_cs:w 434
__phdd_pdfstring_meta:w 434
__phdd_pdfstring_newline:w 415
__phdd_predicate_from_base:n 234, 768
__phdd_print_documented: 1441
__phdd_print_end_definition: 1415, 1441
__phdd_print_macroname:nN 1320, 1324
__phdd_print_testfile:n 1226, 1358
__phdd_print_testfile_aux:n 1358
__phdd_quote_special_char:N 2372, 2429, 2431
\l__phdd_regexes_clist 1786
__phdd_replace_at_at:N . 134, 557, 691, *825*, 2089, 2103, 2125, 2145, 2156
__phdd_replace_at_at_aux:Nn 134
__phdd_shorthand_meta: 934, 937
__phdd_shorthand_meta:w 937
__phdd_show_functions_defined: 2168, 2219
__phdd_show_not_tested: 2234, 2289
__phdd_signature_base_form:n 214, 300
__phdd_signature_base_form_aux:n 214
__phdd_signature_base_form_aux:w 214
__phdd_special_index:nn .. 1033, 1307, 2306, 2329
__phdd_special_index_aux:nnnnnn 2342
__phdd_special_index_module:nnnnN ... 598, 2321, 2334, 2342
__phdd_special_index_set:Nn 2342
__phdd_split_function_auxi:w 240
__phdd_split_function_auxii:w 240
__phdd_split_function_do:nn 240, 268, 305, 705
\l__phdd_store_tl 1733, 1757, 1769, 1782, 1875, 1886, 1905, 1924, 1935, 1945, 1967, 1977, 2010, 2049, 2087, 2088, 2089, 2095, 2101, 2102, 2103, 2109, 2112, 2123, 2124, 2125, 2131
__phdd_str_if_begin:nnTF 124, 198, 201, 206
__phdd_syntax:w . 498, 1101
\g__phdd_syntax_box .. 21, *808*, *836*, 918, 921, 964, 1104, 1112, 1132
\l__phdd_syntax_coffin 20, *807*, *808*, 922, 963, 969, 986, 997
\l__phdd_syntax_dim . 1101, 1106, 1117
__phdd_syntax_end: . 500, 1101
__phdd_test_missing:n 1394, 1395
__phdd_test_missing_aux:Nnn 1397, 1402, 1409
\g__phdd_testfiles_seq 29, *808*, 1363, 1365
__phdd_tmp:w 120, 121
__phdd_tmpa:w 86, 244, 260, 2241, 2258, 2263, 2277, 2470, 2473
\l__phdd_tmpa_int . 49, *810*, 1471, 1473, 1476, 2282, 2285
\l__phdd_tmpa_seq . 50, 175, 176, 177, 806, 807, 808, *810*, 1426, 1430, 1450, 1454, 1456, 1459, 2504, 2508, 2512, 2514
\l__phdd_tmpa_tl .. 47, 266, 268, 609, 615, 618, 678, 682, 685, 687, 688, 689, 690, 691, 693, 746, 747, 750, 751, 807, *810*, 813, *826*, *828*, *831*, 878, 880, 884, 1064, 1065, 1078, 1079, 1332, 1333, 1334, 1339, 1340, 1342, 1404, 1405, 1406, 1407, 1443, 1444, 1446, 1480, 1481, 2143, 2144, 2145, 2149, 2158, 2159, 2238, 2252, 2270, 2280, 2284
__phdd_tmpb:w ... 86, 2245, 2247, 2264, 2265
\l__phdd_tmpb_tl .. 48, 555, 556, 558, 613, 615, 758, 760,

808, *810*, 814, 815, *831*, 2152, 2154, 2156, 2158
\l__phdd_trial_width_dim 24, *808*, 947, 949, 952, 1110
__phdd_trim_right:Nn 118, 179, 181, 185, 186, 187
__phdd_typeset_aux:n 640, 1045, 1349
__phdd_typeset_dates: 1012, 1083
\g__phdd_typeset_documentation_bool 74, 76, 459, 463, 468, 472, *812*
__phdd_typeset_exp: . 517, 640, 1039
__phdd_typeset_expandability: 1024, 1036, 1055
__phdd_typeset_function_block:nN 779, 1017
__phdd_typeset_functions: *835*, 948, 1001
\g__phdd_typeset_implementation_bool 75, 77, 461, 465, 475, 479, *812*, 1595, 1598, 1602, 2171
__phdd_typeset_rexp: 525, 640, 1040
__phdd_typeset_TF: . 533, 640, 1023, 1054, 1353
__phdd_typeset_variant_list:nN 1026, 1042
\l__phdd_undoc_def_tl 2167, 2178, 2197, 2213
\l__phdd_variants_prop 79, *812*
\g__phdd_variants_seq 52, 753, 757, 758, *810*, *830*, 1025, 1047, 1048, 1051, 1275, 1284
__phdd_xmacro_code:n *853*, 1727, 2071, 2079
__phdd_xmacro_code:w 1727, 2082, 2121, 2141
phdd_manual_entry environment 754
\phddcolor 1839
phddoc (class) 390, 681, 721, 726
phddoc commands:
\phddoc_orig_meta ... 1025
\phdidxfont 3166, 3170
\phdindexcom ... 1260, 1383, 1635
\phdindexprintca 1176, 1213, 1229, 1238, 1253
\phdindexprintcolc ... 127, 1315
\phdindexprintcolca . 1251, 1309
\phdindexprintcomc . 1198, 1267, 2516
\phdIndexPrintCs 1799, 1833, 1850
\phdindexprintenvc .. 1216, 1284
\phdindexprintenvca . 1211, 1277
\phdindexprintkeyc . 1232, 1295, 1298, 1496
\phdindexprintkeyca 1227, 1291, 1489
\phdIndexPrintValC ... 170, 1241
\phdindexprintvalca .. 163, 1236
\phdlabelprefix 43, 52
\phdmakedocSubKey 1450
\phdwritetemp 1171
\phimeson 2188
\phimesonnull 2189
\phonon 2190
\photon 2096
\pickup@font 167
picture 333
picture environment 328, 335
\pignenfont 2200
\pionminus 2191
\pionnull 2192
\pionplus 2193
\pkg 452, *804*, 1772, 2359, 2492
\pkgname 783
\pkgname .. 1927, 1928, 1929, 2027, 2053, 2086, 2214, 2316, 2472, 2497, 2513, 2583, 2584, 2591, 2592, 2593
\plasmon 2194
\plotmarkentry 2844
\plotmarkentrytikz 2850
\pmatrix 265
\pmod 263
\polariton 2195
\polaron 2196
\positron 2197
\Pr(Pr) a783
\preface 2
\prefacename 2, 15, 2352
prelim2e (package) 426
prg commands:
\prg_break 276, 278, 282, 288
\prg_break_point 290
\prg_generate_conditional_variant 132
\prg_new_conditional 325, 783
\prg_new_protected_conditional 109, 124, 196
\prg_return_false 114, 130, 208, 210, 346, 350, 355, 788, 796

\prg_return_true . 115, 129, 199, 202, 207, 346, 350, 355, 788, 796
\prgcolor 1823
\prime@s 267
\printanat 2690, 2705
\PrintChanges 191, 192, 200, 1663, 1664, 1675, 2537
\PrintDescribeEnv 1499
\PrintIndex .. 189, 190, 199, 1661, 1662, 1674, 2315
\printindex 389, 745
\printproperties 192
\ProcessKeysPackageOptions 124
\ProcessOptions 512
\ProcessOptions 123, *518*
\ProcessOptions* 500, 512
\ProcessOptionsX 644
prop commands:
\prop_get 106, 1404
\prop_get:NnNTF 88
\prop_gput 108
\prop_gput:Nnn 88
\prop_if_empty 2239
\prop_map_function .. 2257
\prop_new 27, 79
\prop_put 107, 1407
\prop_put:Nnn 88
\propcolor 1825
\protect 52
\protect *382*
\protected@edef 51
\protected@write 97
\protected@xdef 51
\protein 2161
\proton 2162
\provide@command 47
\providecommand 46
\providecommand 159, 161, 163, 449, 450, 451, 1354
\ProvideFile *107*
\provides_module *544*
\ProvidesClass 511
\ProvidesFile 511
\ProvidesFile 486
\ProvidesFile 3
\ProvidesPackage 510
\ProvidesPackage 499
\ProvidesPackage . 2, 6, *111*, 2045, 2046, 2048
\ProvideTextCommand 112
\ProvideTextCommandDefault 114
\ps@empty 409
\ps@plain 409
\PSfont 2527
\put *342*

Q

\Q 778
\qbezier 1
\qbeziermax 1
\Qc 779
\Qc 2328
\qcolor 1831
\Qe 780
\Qe 2343
\Qiv 2357, 2365
\QivBAR 2360, 2366
\Qpc 2351, 2365, 2366, 2373
\Qpfc 2353
\qquad 89
\qquad 2017
\Qt 2349
\quad 89
\quad 89, 231, 250, 251, 252, 253, 254
\quadrupole 2163
Quantum Mechanics
History 389–390
\quark 2164
quark commands:
\q_mark 249, 250, 254, 255, 257, 326, 328
\q_no_value 39, 549, 550, 933, 943, 1224, 1241
\quark_if_no_value 586, 593, 2332
\q_stop 120, 121, 215, 232, 251, 254, 255, 257, 326, 328, 358, 2082, 2121, 2141, 2470, 2473
\quarkb 2165
\quarkc 2166
\quarkcolor 1828
\quarkd 2167
\quarks 2168
\quarkt 2169
\quarku 2170
quotation environment 278
\quotechar 2433, 2436
\Qv 2370, 2373

R

\R 781
\r@@t 264
\raggedbottom 410
\raggedbottom *410*
\raggedleft 1
\RaggedRight 293

\raggedright 1, 225, 457, 1118, 1707, 2960, 3098
\raisebox 516, 524, 532, 2351, 2558
\rbrack 20
\ref 247, 248
\ref *248*, 1534, 1550
ref commands:
\ref_com: 1553, 1566
\ref_com_star . . . 1559, 1566
\refAux 761
\refAux 1591
\refAuxcs 1592
\refCmd 760
\refCmd 1563
\refEnv 1573
refEnv commands:
\refEnv: 1570, 1577
\refEnv_star 1571, 1576
\reference 2171
\refKey 760
\refKey 1583
\refname 2353
\refstepcounter 249
\refstepcounter 60, *247*, 865, 991, 2116, 2119, 2629
regex commands:
\c_bool_regex 1855
\c_box_regex 1860
\c_braces_regex 1907, 1920
\c_brackets_regex 1909, 1941
\c_char_regex 1856
\c_clist_regex 1850
\c_comments_regex . . 2000, 2007
\c_cs_regex 1790, 1794
\c_dim_regex 1853
\c_exp_regex 1852
\c_group_regex 1859
\c_handlers_regex 1776, 1779
\c_ifs_regex 1750, 1754
\c_int_regex 1857
\c_iow_regex 1858
\regex_const . . . 1750, 1776, 1790, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1907, 1908, 1909, 1950, 2000
\c_regex_regex 1863
\regex_replace_all . . 574, 1753, 1765, 1778, 1785, 1871, 1880, 1901, 1918, 1929, 1939, 1961, 1970, 2007
\regex_replace_once . . 869
\c_seq_regex 1849
\c_skip_regex 1861
\c_str_regex 1854
\c_tex_regex 1851
\c_tl_regex 1848
\c_token_regex 1862
regex internal commands:
__regex_compile:n . . 1791
\l__regex_internal_regex 1794
\regexcolor 1814
\registernumber *543*
registers
1,3,5,7,9 10
allocation 10
\relax 1, 2, 3, 154, 172, 188, 190, 192, 196, *214*, 227, 230
remove commands:
remove_from_callback . . . 1
\remove_from_callback *544*
\removeats 2695
\removelastskip 1
\renew@command 43
\renew@environment 45
\renewcommand 43
\renewcommand 20, 286, 347, 411, 438, 454, 463, 485, 492, 512, 517, 1007, 2093, 2094, 2095, 3081, 3091
\renewenvironment 44
\repeat 1
\RequirePackage 514
\RequirePackage 486
\RequirePackage 2, 3, 5, 27, 130, 131, 132, 134, 135, 136, 137, 1003, 1392, 1727, 2057
\RequirePackages *493*
\RequirePackageWithOptions 515
\resistivity 2172
\restore@protect 51
\restorecr 89
\restoresymbol 2323
\returntospace 2878, 2932
\reversemarginpar 1, *378*
\rhomesonminus 2173
\rhomesonnull 2174
\rhomesonplus 2175
\rightarrow 2225
\rightline 1
\rightmargin 287
\rightmargin 1337
\rightmark 1
\rightskip *284*
\Righttorque . 243, 272, 366, 1016

\rlap 1
\rmfamily 1031, 1039
Robertson, Will 207
\Roman 148
\roman 148
\romannumeral 1212, 1218
\root 264, 270
\rootbox 264
\rq 1
\rule 1935

S

\samepage 80
\samepage 3032
\sanitize 766
\sanitize . 1790, 1812, 1823, 1839, 1857, 1868, 1879
\sanitized 1790, 1796, 1799, 1805, 1812, 1815, 1818, 1823, 1826, 1829, 1839, 1850, 1857, 1860, 1863, 1868, 1871, 1874, 1879, 1882, 1885
\sarrow 2780
\savebox 306
\savebox 2557
\savesymbol 2321
\sb 266
\sb 622
\sbox 307
\sbox *24*
scan commands:
\scan_stop: 127, 128, 189, 714, 724, 751, *810*, 814, *829*, 1528
\scantokens 1834, 2860, 2937
scriptexample environment . . 728
scripts:
cypriote 764
\scriptsize 1
\searrow 2226
\sec(sec) a783
\secdef 1
\section 914
section commands:
\section_cntformat . . . 256
\section_number_after_tl 247, 250, 256
\section_number_after_tl . 910
\sectionmark 1
\sectionname 885
sectsty (package) 910
\seename 2369
\selectfont *161*
seq commands:
\seq_clear 681
\seq_clear_new 734
\seq_count . 1047, 1454, 1472, 2508
\seq_gclear 753, 1484
\seq_get 750, 808
\seq_gpop 758
\seq_gput_right . . 89, 1032, 1248, 1253, 1303, 1365, 1428
\seq_gset_eq 757
\seq_gset_filter 1465
\seq_if_empty . . 1025, 1275, 1469, 2261
\seq_if_in . 1363, 2181, 2195
\seq_item . . 1459, 1479, 2512, 2514
\seq_map_function 699, 2276
\seq_map_function:NN . *841*
\seq_map_inline 1284, 2179, 2193
\seq_map_variable 177
\seq_new 11, 12, 28, 29, 52, 54, 55
\seq_pop 746, 807
\seq_pop_left 176
\seq_put_right 682, 735, 737
\seq_set_eq 806
\seq_set_filter 1426
\seq_set_from_clist . . 692
\seq_set_map 1450
\seq_set_split 88, 175, 2504
\seq_set_split:Nnn . . . 88
\seq_use . . . 1048, 1051, 1399, 1430, 1456
\seq_use:Nnnn *847*
\seqcolor 1815
set commands:
\set_borderline_aux:nn 21, 210, 254
\set_borders_aux:nn 11, 26, 32, 38, 42, 218, 219, 220, 221
\set_color:nn 335, 369
\set_color_aux:nn . 15, 216, 217
\set_font_aux:n 2, 67, 78, 102, 107, 126, 148, 272, 294, 326, 328, 336, 351, 355, 362, 370, 422, 447, 471, 499, 525
\set@display@protect 51
\set@typeset@protect 51
\set_color_aux:nn 894
\set_font_aux:n 894
\setattribute *542*
\setboolean 486

\setcaptions 4, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121
\setcounter 146
\setcounter 152
\sethyperref 133
\setkeys 2912
\setlength 153
\setlength 148
\setrefcountdefault 1526
\SetSymbolFont 2092, 2094, 2102
\setto 154
\settodepth 1945
\settoheight 154
\settoheight 1944
\settowidth 154
\sffamily .. 26, 46, 405, 3154, 3175
\sh@ft 25
\shipout 421, 455
\shipout *98, 422, 423, 425*
\shortstack 335
\showaccent 801
\showaccent 3230
\showhyphens 174
\showlbrace 800
\showlbrace 3215
\showmbrace 800
\showmbrace 3210
\showop 800
\showop 3206
\showoutput 1
\showover 3234
\showoverfull 1
\showrbrace 801
\showrbrace 3221
\showrelsymbol 800
\showrelsymbol 3200
\showrover 801
\showsymbol 800
\showsymbol 3194
\showsymbolalpha 799
\showsymbolalpha 3187
\showsymbolbin 800
\showsymbolbin 3197
\showwideaccent 3226
\sigma 1701
\sin(sin) a783
\sinh(sinh) a783
\skewchar 141
skip commands:
\skip_gset:Nn 11
\skip_vertical:N .. 89, 136
\l_tmpa_skip 7, 9, 11, 24, 174, 175, 177, 197, 198, 211
\slash 22
\SLiTeX *91*
\sloppy 410
sloppypar (environment) 1
\smallbreak 1
\smallskip 86
\smallskip . 1679, 2715, 2716, 3163
\smallskipamount 87
\smallverbatim 3132
\smallverbatimsize 3134
\smash 265
\solid 2176
\sourceformat 660
\sp 266
\space 20
\space 179
spacing
\hglue 22
\vglue 22
french 19
non-french 19
``special'' characters .. 772
\SpecialIndex 2303
\SpecialMainEnvIndex ... 1636
\SpecialMainIndex .. 50, 1635
\SpecialOptionIndex ... 1501, 1505
\SpecialPackageIndex 784
\SpecialPackageIndex ... 295, 2478, 2494
\spin 2177
\spindown 2178
\spinup 2179
\split@name 168
\splittopskip 419
\sqrt 270
\sqrtsign 270
\sqrtsign 270
\Square() a772
square commands:
\c_square_brackets_regex 1908, 1931
\squarebracketscolor ... 1913
\ss 674
\stackrel 269
\star 642
start commands:
\start_section:nnnnnnnnn ... 2, 48, 289, 387

\start_section:nnnnnnnnn 904
\startlineat 731
\startlineat 331, 332
\startnumberat 333
\stepcounter 147
\stewart 121
\stok 721, 723
\stok 3180
\stop 1
\StopEventually 1594
str commands:
\c_backslash_str ... 412, 424, 430, 589, 672, 813, 2413, 2426, 2450
\c_percent_str 686
\str_case 90, 218, 2377, 2494
\str_case:nn 88
\str_case_e 85
\str_case_x 91
\str_count 1337
\str_head 2494
\str_if_eq 298, 751, 814, 2352, 2374
\str_tail 2492
\c_underscore_str ... 671
\strcolor 1829
\stretch 89
\string ... 14, 94, 111, 153, 179, 183, 208
\stripcommand 2697, 2701
\strippedat 2688, 2690, 2696
\strippedtext 2697, 2698
\strut 1, 224
\strutbox 1
\subitem 590
\subparagraph 916
\subparagraph 512, 517
subparagraph commands:
\subparagraph_cntformat 260
\subparagraph_number_after_tl 254, 260
\subparagraphmark 1
\subsection 914
\subsection ... 12, 438, 454, 1542, 1558, 2208
subsection commands:
\subsection_cntformat 257
\subsection_number_after_tl 251, 257
\subsectionmark 1
\subsubitem 590
\subsubsection 915
\subsubsection 14, 463, 1550, 1559, 1576, 1582, 1583, 1589
subsubsection commands:
\subsubsection_cntformat 258
\subsubsection_number_after_tl 252, 258
\subsubsectionmark 1
\sup(sup) a783
\supereject 23
\suppressfloats 473
\surface 2180
sverbatim environment ... 730
\SVR 2086, 2146
\svrexample 2181
\svrphoton 2182
\SVRtrue 2089
\swarrow 2227
\symarrow 2766
\symbola 203
symbols
currency 765, 788
log-like 783
monetary 765, 788
musical 779
variable-sized 781
\symbolsdecimal 43
\symbolsexponential 55
\symbolsgroup 45
\symbolsinfinity 61
\symbolslist 47
\symbolsminussign 53
\symbolsnan 59, 63
\symbolspermille 49
\symbolsplussign 51
\symbolssuperscriptingexponent 57
\symbolstimeseparator 65
syntax (environment) ... 497, 804
sys commands:
\c_sys_jobname_str 2174, 2313, 2534, 2535
\syscolor 1843

T

\tabbing 1590
\tabcolsep 1008
\tablename 2124, 2363, 2619, 2632
\tableofcontents 587
\tachyon 2183

\tan(tan) a783
\tanh(tanh) a783
\task 168, 242
\tauleptonminus 2184
\tauleptonplus 2185
\TC 2576
\tcbdimto 1365, 1366
\tcbdocmarginnote 54
\tcbIndexPrintComC 114
\tcblisting .. 950, 952, 955, 958, 994, 996
\tcblower 1354
\tcbox .. 109, 148, 154, 174, 191, 198
\tcbset 23, 29, 38, 44, 55, 186, 788, 791, 871, 964, 1159, 1160, 1644
\tcbuselibrary 28
\Tding 2408
\TemplateArgument 1546
\TemplateKey 1563, 1566
\TemplateSemantics 1547, 1573
\tencirc 3
\tencircw 3
\tenln 2
\tenlnw 2
\testfile 1226, 1392
\TestFiles 806
\TestFiles 806, 1382
\TestMissing 806
\TestMissing 806, 1393
\TestSections 18
\testsections 926
\testsections 8, 18
\TeX ... 1, 90, 92, 104, 145, 152, 157, 176
\tex 1080
TEX and LATEX 2ε commands:
\@ 515, 825
\@@@@par 3118, 3125
\@@endpbox 1
\@@input 55
\@@line 1
\@@par 284
\@@startpbox 1
\@DEBUGfalse 13, 38
\@DEBUGtrue 18
\@M 1, 86, 133, 138, 188, 223, 325, 350, 1544, 1561, 1585
\@MM 1, 57
\@Mi 1
\@Mii 1
\@Miii 1
\@Roman 316, 330
\@aalph 211, 212
\@addtoreset .. 165, 166, 402, 783, 786
\@afterheading 1
\@afterindentfalse .. 12, 17
\@afterindenttrue . 1, 8, 18
\@arabic . 103, 337, 344, 368, 375, 382, 389, 396, 790
\@author 1
\@auxout .. 93, 178, 186, 205, 247, 1640, 1658, 1684
\@begin@tempboxa 1
\@begintheorem 1
\@bezier 1
\@bitor 437
\@botlist 435
\@car 248
\@carcube 1
\@cclv 1
\@cclvi 1
\@cdr 248
\@centercr 1
\@checkend 1
\@circ 1
\@circle 1
\@clnht 1
\@clnwd 1
\@clubpenalty .. 200, 232
\@combinefloats 435
\@compatibility 517
\@ctrerr 218
\@currdir 7
\@currentlabel . 51, 247, 1384, 1409, 1432, 1438
\@currenvir ... 522, 2374
\@currenvline 1
\@dashbox 1
\@dashcnt 1
\@dashdim 1
\@date 1
\@dbflt 1
\@dblfloat 1
\@dblfloatplacement .. 1, 432
\@declaredoptions ... 518
\@dischyph 564

\@docinclude .. 180, 182, 1642, 1644
\@documentclasshook . 517
\@dot 1
\@dottedtocline 1
\@downline 1
\@downvector 1
\@eha 179, 1641
\@empty 210, 256
\@end@tempboxa 1
\@endpbox 1
\@endtheorem 1
\@evenfoot 226, 241, 1710, 1725
\@filelist 1
\@filesw 93, 94
\@finalstrut 1
\@floatboxreset 1
\@floatpenalty 1
\@flushglue 1
\@footnotemark 1
\@footnotetext 382
\@for 184, 1652
\@framebox 1
\@freelist 435
\@getcirc 1
\@getlarrow 1
\@getrarrow 1
\@gobble 221
\@gobble@till@return 2884, 2885, 2899
\@gobble@till@return@ignore 2886, 2899
\@halfwidth 1
\@hangfrom 1, 6
\@highpenalty 1
\@hline 1
\@hvector 1
\@icentercr 1
\@iden 1
\@idxitem ... 1610, 1611, 2759, 2760
\@ifnextchar 1966, 2253, 2289, 2293, 2433, 2452
\@ifpackagelater 506
\@iframebox 1
\@ignorefalse 1
\@ignoretrue 1
\@iiiminipage 1
\@iiminipage 1
\@imakebox 1
\@imakepicbox 1
\@iminipage 1
\@index 1
\@indexfile ... 207, 1688
\@input 93, 94, 183
\@inputcheck 37
\@ishortstack 1, 1
\@istackcr 1
\@ixstackcr 1
\@keyval 3017, 3020
\@killglue 1
\@largefloatcheck 1
\@latexerr 25, 179, 1641
\@leftmarginskip 378
\@leftmark 1
\@linechar 1
\@linelen 1
\@lowpenalty 1, 76
\@ltxcompatfalse .. 7, 45
\@ltxdoc@PrintChanges 1663, 1675
\@ltxdoc@PrintIndex 1661, 1674
\@ltxdoc@endtheglossary 1666, 1677
\@ltxdoc@theglossary 1665, 1676
\@m 1, 57
\@mainaux .. 93, 94, 183, 205, 1647, 1684
\@mainout 93
\@makebox 1
\@makecaption 369, 2125, 2633
\@makefnmark 1
\@makeother 1
\@makepicbox 1
\@makeschapterhead . 1627
\@marginparreset 1
\@markright 1
\@medpenalty 1
\@minipagerestore 1
\@minus 2550, 2551, 2552, 3077, 3087
\@miv 1
\@mparswitch 378
\@mpfn 1, 382
\@mpfootins 1
\@mplistdepth 1
\@multicnt 1
\@nameuse 205, 1682
\@ne 1, 57
\@newctr 1
\@nil 248
\@nmbrlist 284
\@nobreakfalse . 182, 222
\@nobreaktrue 218

\@nodocument 96
\@noligs 1, 259
\@nomath 1
\@noskipsecfalse 186
\@noskipsectrue .. 1, 183
\@nthm 1
\@oddfoot 224, 226, 229, 241, 1702, 1710, 1713, 1725
\@onlypreamble 249
\@opargbegintheorem ... 1
\@opcol 435
\@othm 1
\@outputbox 435
\@outputpage 434
\@oval 1
\@ovdx 1
\@ovdy 1
\@ovhorz 1
\@ovri 1
\@ovro 1
\@ovvert 1
\@ovxx 1
\@ovyy 1
\@partaux . 93, 178, 186, 187, 188, 204, 1640, 1658, 1659, 1660, 1680
\@partlist . 93, 94, 185, 1652
\@partout 93
\@partsw 93, 94
\@phddoc@PrintChanges 191, 200
\@phddoc@PrintIndex 189, 199
\@phddoc@endtheglossary 194, 202
\@phddoc@theglossary 193, 201
\@picbox 1
\@picht 1
\@picture 1
\@picture@warn 1
\@plus 1011, 1012, 2550, 2551, 2552, 3077, 3087
\@providesfile 1
\@ptsize 1
\@put 1
\@restonecoltrue .. 1600, 2750
\@rightmark 1
\@rightskip 1
\@roman 309, 323
\@savemarbox 1
\@seccntformat ... 1, 65, 244
\@secpenalty 1, 23
\@sect 1
\@sectionspaceoutfalse 511
\@sectionspaceouttrue 510
\@setfps 1
\@setminipage 1
\@setnobreak 1
\@setpar 284
\@sline 1
\@smallverbatim ... 3114, 3132
\@ssect 1, 37, 131
\@stackcr 1
\@startcolumn 435
\@startpbox 1
\@startsection 1, 48, 412, 440, 454, 465, 485, 493, 512, 519
\@starttoc 1
\@strip@args 1
\@svector 1
\@sverb 1
\@svsec . 57, 57, 63, 198
\@svsechd 157
\@sxverbatim 1
\@tempa 185
\@tempb 184, 185
\@tempboxa 1
\@tempboxb 100, 106
\@tempcnta 1
\@tempcntb 1
\@tempdima 1
\@tempdimb 1
\@tempdimc 1
\@tempskipa . 1, 81, 132, 133
\@tempskipb 1
\@tempswafalse 184
\@tempswatrue .. 184, 185
\@temptokena 1
\@test... 440
\@testdef 1
\@text@composite 1
\@text@composite@x 1
\@textsuperscript 1
\@tf@r 1
\@thanks 1
\@thefnmark 382
\@themark 1
\@thispagestyle 434

\@thm 1
\@thmcounter 1
\@thmcountersep 1
\@title 1
\@toplist 435
\@topnum 432
\@toproom 432
\@tryfcolumn 434
\@typeset@till@return 2887, 2888
\@undefined ... 781, 2860
\@unknownversion ... 517, 518
\@upline 1
\@upvector 1
\@verb 1
\@verbatim 1
\@vline 1
\@vobeyspaces 1
\@vvector 1
\@whilenoop 60
\@wholewidth 1
\@wrglossary 1
\@wrindex 1
\@writeckpt 93, 204, 1679
\@writefile 1
\@xarg 1
\@xcentercr 1
\@xdblfloat 1
\@xdim 1
\@xfloat 1
\@xfootnote 1
\@xfootnotenext 1
\@xmpar 1
\@xnthm 1
\@xobeysp ... 1, 88, 580, 825
\@xsect 1
\@xsmallverbatim ... 3132
\@xthm 1
\@xverbatim 1
\@xxxii 1
\@xympar 1, 379
\@yarg 1
\@ympar 1
\@ynthm 1
\@ythm 1
\@yyarg 1
\normalsize 487
\setfontsize 487
\sg 495
\@afterheading 577
\@car 39
\@carcube 39
\@cdr 39
\@cons 38
\@height 34
\@mparswitchfalse ... 634
\BaseDirectory 494
\alphabet 197
\begin 635
\cmd 803
\foo 803
\generate 495
\index 389, 390
\multiput 330
\mylist 38
\newif 46
\normalsize 487
\put 330
\string 45
\bovedisplayshortskip 487
\bovedisplayskip 487
\aselineskip 487
\elowdisplayshortskip 487
\ewboolean 486
\ormalsize 487
\ize 487
\mall 487
\sepackage 486
_ 847
\alloc@ 1
\bm@b 1
\bm@c 1
\bm@l 1
\bm@r 1
\bm@s 1
\bm@t 1
\bottomrule 830, 836
\c@bottomnumber 368
\c@chapter 781
\c@CodelineNo .. 85, 209, 1306, 1308, 1691
\c@dbltopnumber 368
\c@HD@hypercount .. 1006, 1513, 2394, 2484, 2810, 2830
\c@IndexColumns ... 1597, 2748, 2749
\c@page 280
\c@secnumdepth ... 1, 55, 329, 354
\c@section 790
\c@tocdepth 1
\c@topnumber 368

\c@totalnumber 368
\calculate@math@sizes . 1
\check@checksum 1605
\circle 342
\cmd@to@cs 153, 154, 1889, 1902, 3054, 3056
\code@catcode@hook 2900, 2903, 2923
\code@end ... 2945, 2956, 2970, 2993
\code@executefalse . 2895
\code@executetrue .. 2910
\code@flushend 2942, 2953, 2963
\code@flushstart .. 2942, 2952, 2963
\code@mid ... 2943, 2954, 2968, 2972
\code@post .. 2897, 2898, 2902, 2937
\code@pre ... 2896, 2898, 2901, 2937
\code@start . 2941, 2947, 2958
\code@temp ... 2934, 2937
\code@width . 2944, 2955, 2969, 2974
\codeline@wrindex .. 206, 1686, 2375, 2379
\color@begingroup 1
\color@endbox 1
\color@endgroup 1
\color@hbox 1
\color@setgroup 1
\color@vbox 1
\count@ 1
\cp@FOO 93
\curr@math@size 1
\DeclareFontShape@ 1
\default@ds 518
\define@key . 2891, 2892, 2893, 2894, 2895, 2896, 2897, 2898, 2899, 2900
\Describe@Option .. 1492, 1494
\dimen@ 1
\dimen@i 1
\dimen@ii 1
\do@noligs 1
\DocInput 851, 851
\document@default@language 1
\docValue@ 156
\docValue@star 156
\DoNotIndex 821
\ds@ 518
\e@alloc 1
\e@alloc@chardef 1
\e@alloc@top 1
\e@ch@ck 1
\end@dblfloat 1
\endtrivlist 844
\expanded@notin 411
\ext@figure␣=␣'lof' . 369
\ext@TYPE 369
\extract@alph@from@version 1
\f@encoding 1
\f@family 561
\filec@ntents 522
\find@example 2925
\first@arg .. 1759, 1760, 1769, 1770, 1791, 1910, 1912, 2661, 2662
\fnum@ 369
\fnum@table . 2125, 2619, 2633
\fnum@TYPE 369
\fps@ 369
\fps@dbl 1
\frozen@everydisplay .. 1
\frozen@everymath 1
\ftype@ 369
\g@addto@macro 223, 2290
\G@refundefinedfalse 248
\HD@savedestfalse .. 1004
\HD@target .. 1005, 1234, 1509, 2382, 2481, 2807, 2827
\HDorg@encapchar .. 1510, 1513, 2482, 2484, 2808, 2810, 2828, 2830
\HDorg@theCodelineNo 13, 404
\hmode@bgroup 129
\Hy@MakeCurrentHref 1006
\idx@actual ... 114, 162, 169, 1082, 1266, 1276, 1283, 1290, 1294, 1297, 1308, 1315, 1489, 1495, 1797, 1805, 1842, 1849, 1892, 1895, 1900, 1902, 2516, 2522, 2523, 2767, 2768, 2812, 2832, 2833
idx@actual 1080
\idx@encap 1080
\idx@format .. 1080, 1146

\idx@level 168, 294, 1080, 1282, 1293, 1296, 1314, 1493, 1717, 1722, 1734, 1747, 1749, 1750, 1751, 1815, 1842, 1847, 1899, 2523, 2768, 2833, 3167
\idx@quote 1080
\if@afterindent .. 1, 224
\if@DEBUG 12, 19, 39
\if@filesw 183, 186, 204, 206, 400
\if@ignore 1
\if@ltxcompat 7, 46, 409, 435, 461, 484, 490, 511
\if@mainmatter 1
\if@minipage 1
\if@negarg 1
\if@newlist 3133
\if@nobreak 20, 221
\if@noskipsec . 1, 4, 185
\if@ovb 1
\if@ovhline 1
\if@ovl 1
\if@ovr 1
\if@ovt 1
\if@ovvline 1
\if@partsw 184, 1649
\if@tempswa 1, 186
\if@twoside 280
\ifcode@execute ... 2906, 2928
\ifnot@excluded 409
\ifphd@doc@annotate 1051, 1183
\ifphd@doc@colorize 1050, 1177, 1190
\ifphd@doc@toindex . 159, 1049, 1262, 1273, 1290, 1304, 1488
\ifwelsh@first 2202, 2211
\ifwelsh@vowel 2212, 2222
\include 851
\index@excludelist . 413, 821
\index@level 2530
\index@prologue 223, 1603, 1617, 1626, 1700, 2742, 2752
\init@checksum 1606
\input 851
\insc@unt 1
\it@is@a 866, 2317
\K@no@opt@arg 1996, 2003
\K@opt@arg ... 1990, 2003
\Kcyp@no@opt@arg .. 1965, 1966
\Kcyp@opt@arg 1964, 1966
\Kgr@no@opt@arg ... 2251, 2253
\Kgr@opt@arg . 2250, 2253
\kvtcb@col@command ... 60
\kvtcb@col@counter .. 288
\kvtcb@doc@body@command@after 56
\kvtcb@doc@body@command@before 55
\kvtcb@index@command 112
\l@nohyphenation 634
\LT@array 2120
\lt@indexed . 2126, 2141, 2142
\LT@start 2121
\ltx@sh@ft 129
\m@ne 57
\m@th 1, 91, 91
\macro@namepart 412, 821
\math@bgroup 1
\math@egroup 1
\meta 826
\meta@font@select .. 630, 1031, 1039
\meta@hyphen@restore 631, 636
\midrule 836
\N@no@opt@arg 2266, 2275
\N@opt@arg ... 2259, 2276
\nfss@text 1, 628
\Np@no@opt@args ... 2292, 2293
\Np@one@opt@arg ... 2289, 2291
\Np@opt@args . 2289, 2293
\Np@two@opt@args .. 2289, 2290
\ORDINALS@cx 396
\ordinals@cx 389
\origLT@array 2120
\origLT@start 2121
\outer@nobreak 1
\oval 342
\p@cnt 247
\p@enum 298
\part 866, 872
\patch@level 1

\phd@doc@head 757
\phd@doc@index@doc . 1633
\phd@doc@index@off . 1642
\phd@doc@index@pgf . 1632
\phd@doc@index@pgf@ 1595, 1623, 1629
\phd@doc@index@pgfchapter 1625
\phd@doc@index@pgfsection 1616, 1632
\phd@doc@orig@meta 1026, 1032, 1040
\phd@doc@toindexfalse ... 1394, 1420, 1447, 1459
\phd@key@path 1453, 1455, 1457
\phd@layer@pushup .. 1171
\phd@print@LstStyle 287, 292
\phd@Print@Val 176, 177, 1246
\phd@text@lststyles 294, 295, 1069
\phdkv@afterexample .. 80
\phdkv@beforeexample . 79
\phdkv@col@opt . 74, 1008
\phdkv@col@value 67, 1238, 1243, 1248
\phdkv@colhyper . 75, 76, 1148, 1149, 1150, 1151, 1152, 1591, 1592
\phdkv@doc@description 1123, 1352, 1355
\phdkv@doc@indentleft 1115, 1365
\phdkv@doc@indentright 1116, 1366
\phdkv@doc@left ... 1113, 1333
\phdkv@doc@right .. 1114, 1337
\phdkv@doclistingoptions 1075
\phdkv@docmintoptions 1078
\phdkv@docmintstyle 1077
\phdkv@listingencoding 1022
\phdkv@text@color . 1057, 1253
\phdkv@text@colors 1058, 1314
\phdkv@text@env ... 1060, 1213
\phdkv@text@envcontent 1059, 1410
\phdkv@text@envs .. 1061, 1281
\phdkv@text@index . 1064, 1619, 1620, 1627, 1628
\phdkv@text@key ... 1062, 1229
\phdkv@text@keys .. 1063, 1292, 1492
\phdkv@text@pageshort 1055, 1065, 1540
\phdkv@text@value . 1066, 1238
\phdkv@text@values . 167, 1067
\pictur@ 1
\pkdkv@colmeta 77
\protected@edef . 51, 63, 1384, 1409, 1432, 1438
\protected@xdef 358
\Q@no@opt@arg 2302, 2311
\Q@opt@arg 2295
\r@foo 248
\R@no@opt@arg 2382, 2391
\R@opt@arg ... 2376, 2392
\refKey@ 1581, 1587
\refKey@star . 1582, 1586
\rem@pt 1
\saved@indexname ... 843, 1310
\saved@macroname ... 843, 1290
\second@arg . 1911, 1913, 1919
\set@color 1
\setfont@cx 298, 397
\setfontparam@cx ... 303, 402, 427, 448, 476, 504, 526
\sh@ft 129
\sixt@@n 1
\sixt@@n␣ 57
\skip@ 1
\smallverbatim@font 3128, 3134
\strip@pt 1
\tcb@cs 48, 51
\tcb@doc@bfseries ... 288
\tcb@doc@do@description 54
\tcb@doc@toindexfalse 58
\tcb@index@Com .. 49, 124
\tcb@Print@Com ... 47, 60

\tcb@scantokens 288
\tcb@temp@grow@left 1365, 1367
\tcb@temp@grow@right 1366, 1368
\tcb@w@upper@real .. 1371
\telugu@month 2022, 2036
\temp@cx 3013, 3021, 3022, 3024
\texttt 846, 847
\thr@@ 1
\toks@ 1
\toprule 836
\trivlist 844
\tw@ 1, 24, 57
\verb 867
\verb@balance@group ... 1
\verb@egroup 1
\verb@eol@error 1
\verbatim@ ... 2135, 2642
\verbatim@addtoline 2132, 2639
\verbatim@finish .. 2134, 2641
\verbatim@font 1, 20, 259, 567, 825, 867, 2390
\verbatim@nolig@list .. 1
\verbatim@processline 2133, 2640
\verbatim@startline 2131, 2638
\voidb@x 1
\W@no@opt@arg 2429, 2433
\W@opt@arg ... 2425, 2433
\welsh@article 2204, 2265
\welsh@article@do . 2207, 2210
\welsh@firstfalse .. 2218
\welsh@firsttrue ... 2205
\welsh@isvowel 2211, 2223
\welsh@ordinal@long 2232, 2260, 2265
\welsh@ordinalplusxx@long 2255, 2259
\welsh@vowelfalse .. 2214
\welsh@voweltrue .. 2224, 2227, 2229
\WORDS@cx 382
\Words@cx 375
\words@cx 368
\X@no@opt@arg 2441, 2446, 2451, 2452
\X@opt@arg ... 2450, 2452
\xmacro@code . 853, 1727, 2070
\z@ 1, 57
\z@skip 1, 560, 564, 3116, 3117
tex commands:
\tex_clubpenalty ... 188, 200
\tex_day 880
\tex_everypar . 184, 201, 219, 233
\tex_ignorespaces ... 92, 205
\tex_interactionmode 2282, 2285
\tex_lowercase 2067
\tex_month 880
\tex_par 134, 339
\tex_unskip 196
\tex_year 880
\texcolor 1840
texexample environment . 741, 744
\texexample 962
texnote (environment) .. 503, 805
\text 2417
\textbackslash 48, 437, 3060
\textbaht() a765
\textbf 507, 1586, 3175
\textcent(¢) a765
\textcolonmonetary(₡) .. a765
\textcolor 60, 135, 156, 288, 340, 405, 1007, 1008, 1178, 1191, 1207, 1224, 1248, 1483, 1591, 1592, 1911, 1913, 1935, 2534, 3058, 3101, 3103
textcomp (package) 765
\textcurrency(¤) a765
\textcypr 1964, 1965
\textdir ... 1398, 1400, 1415
\textdollar($) a765
\textdong(₫) a765
\texteuro(€) a765
\textfloatsep 481
\textfloatsep 435
\textflorin(ƒ) a765
\textfraction 481
\textfraction 368
\textguarani(₲) a765

\textindent 24
\textit ... 1375, 1385, 2154, 2158, 2715, 2716, 3154
\textlira(£) a765
\textls 297
\textnaira(₦) a765
\textoken 2711
\TextOrMath 150
\textpeso(₱) a765
\textquotesingle 676
\textrm 241
\textrm 1050, 1051, 1052
\textsc 311, 312
\textsf 499
\textsf 455, 456, 1378, 1388, 2505, 2506, 2507
\textsl 297, 2888
\textsterling(£) a765
\textsuperscript 1, 631, 1537
\TextSymbolUnavailable 112
\texttt . 157, 448, 450, 451, 453, 518, 520, 527, 528, 533, 534, 1024, 1179, 1181, 1191, 1192, 1446, 1534, 1550, 1721, 1748, 1757, 1761, 1763, 1771, 1773, 2361, 2474, 2475, 2499, 2501, 2510, 2511, 2515, 2521, 2522, 2523, 2544, 2550, 2551, 2552, 2663, 2666, 2714, 2720, 2721, 2724, 2725, 2726, 2737, 2738, 2767, 2768, 2769, 2781, 2782, 2783, 2784, 2785, 2797, 2798, 2799, 2812, 2815, 2816, 2832, 2833, 2845, 2846, 2847, 2851, 2852, 2853, 3103, 3181, 3190, 3195, 3198, 3201, 3204, 3208, 3212, 3218, 3223, 3227, 3231, 3236
\textunderscore 562
\textvcenter 2556
\textwidth ... 148, 212, 224, 682, 956, 1108, 1704, 2565, 2948
\textwon() a765
\textyen(¥) a765
teXX environment 731
teXXX environment 732
\thanks 499
\thanks 357
\thanks 1, 358
\the 499
thebibliography environment 589
thebibliography (environment) 403
\thechapter 82
\theCNT 247
\theCodelineNo 12, 403
thecs a726
\theenum 298
\thefigure 369
\thefootnote 1
\theglossary 193, 201, 1665, 1676
\theindex .. 1598, 2290, 2750
\themeauthor 2716
themeta a746
\thempfn 1, 382
\thempfootnote 1
theoption a797
\thepage 239, 1535, 1723
\theparagraph 259, 317
\thepart 197, 220, 1671, 1697, 1698, 1716, 1719
thepart commands:
\thepart: 231, 234
\thesection 144, 256
\thesubparagraph 260
\thesubsection 184, 257
\thesubsubsection .. 226, 258
\thetable 2619
thetable commands:
\thetable: ... 2124, 2632
\thetexexp 790, 866, 992
\thetotalsymbols 1933
\thicklines 1, 329
\thickmuskip 231
\thinlines 1, 329
\thinmuskip 231
\thinspace 89
\thinspace 1242, 1243, 1315, 3021, 3025
\thirdlist 111, 115
\thispagestyle 1, 434
\tikz 2848, 2853
\tikzi .. 10, 20, 21, 33, 40, 53, 201, 227, 235, 282, 313, 322, 331, 339, 352, 361, 370, 378
\tiny 1, 362, 405, 662, 2351
\title 357
\title 1

tl commands:
\c_empty_tl 871
\c_space_tl . 1400, 2390, 2427
\tl_clear 698, 1225, 2176, 2177, 2178, 2231, 2238, 2444
\tl_const 61, 63
\tl_count .. 91, 112, 113
\tl_count:n 88
\tl_gclear ... 1617, 1694
\tl_gput_right .. 62, 64, 1700, 2315, 2537
\tl_greplace_all 92, 748
\tl_greplace_all:Nnn . 88
\tl_gset 72, 1605, 2087, 2101, 2123, 2148
\tl_gset_eq 747
\tl_if_empty ... 93, 136, 331, 332, 333, 335, 336, 337, 393, 791, 1089, 1095, 1245, 2222, 2280, 2349, 2385, 2502
\tl_if_empty_p 1086, 1087
\tl_if_eq 1361
\tl_if_head_eq_charcode . . 94, 95, 96, 312, 713, 2449, 2477, 2482, 2491
\tl_if_head_eq_charcode:nNTF 88
\tl_if_head_eq_meaning 97
\tl_if_head_eq_meaning:nNTF 88
\tl_if_in 98, 99, 100, 101, 126, 275, 277, 279, 284, 731, 2081, 2084, 2120, 2445, 2452, 2455, 2458
\tl_if_in:NnTF 88
\tl_if_in:nnTF 88
\tl_map_function 741
\tl_map_inline 610, 803, 2408, 2433
\tl_map_inline:Nn ... 830
\tl_new 15, 16, 38, 40, 47, 48, 51, 56, 57, 58, 60, 60, 68, 69, 69, 70, 78, 82, 83, 84, 1614, 1733, 1810, 2165, 2166, 2167, 2342, 2343
\tl_put_left 673
\tl_put_right . 281, 286, 733, 1405, 1797, 1811, 2149, 2158, 2183, 2188, 2197, 2252, 2270
\tl_remove_all 102, 672, 685, 687, 688, 689
\tl_remove_all:Nn 88
\tl_replace_all 103, 144, 145, 146, 147, 148, 149, 150, 151, 153, 154, 155, 156, 157, 158, 159, 160, 162, 174, 556, 558, 580, 615, 671, 1340, 1444, 2402, 2411, 2425, 2435
\tl_replace_all:Nnn .. 88
\tl_replace_once 12, 104, 287, 403
\tl_replace_once:Nnn . 88
\tl_set ... 39, 121, 250, 251, 252, 253, 254, 267, 270, 274, 549, 550, 552, 570, 588, 595, 609, 670, 678, 868, 873, 893, 933, 943, 1202, 1224, 1241, 1290, 1310, 1339, 1406, 1443, 1697, 1698, 1794, 2143, 2152, 2371, 2399, 2417, 2424, 2442, 2459, 2465, 2472, 2496, 2497, 2506
\tl_set_eq 173, 184, 2333, 2485, 2519
\tl_set_rescan . 555, 613
\tl_tail 2466
\tl_to_str 48, 105, 113, 127, 128, 189, 190, 191, 237, 249, 274, 275, 277, 279, 281, 284, 287, 360, 361, 412, 413, 437, 439, 570, 596, 670, 678, 687, 693, 794, 800, 883, 1020, 1076, 1294, 1330, 1708, 2399, 2402, 2409, 2436, 2443, 2445
\tl_to_str:n ... 88, 814, 817
\tl_use 1613, 1846, 2095, 2109, 2112, 2131, 2159
\tlcolor 1816

\Tm 2413
\Tmesonminus 2186
\Tmesonnull 2187
\Tmesonplus 2188
\Tmoon 2414
\tn . 420, 444, 803, 812, 825
\today 591
\today 1, 7
token commands:
\token_to_str . 144, 145, 146, 147, 153, 154, 155, 156, 162, 174, 187, 260, 297, 421, 435, 1063, 1249, 1445, 2334, 2336, 2337, 2338, 2340, 2390, 2405, 2406, 2409, 2414, 2419, 2427, 2452, 2455, 2458, 2486, 2487, 2505, 2520
\tokencolor 1830
\topfigrule 483
\topfraction 480
\topfraction 368
\topinsert 423
\TOPLIST 435
\topmark 420
\toprule 1116, 2349
\topsep 274, 282
\topskip 287
\topskip 1, 96
totalsymbols counter 770
\tracingfonts 1
\tracingnone 1
\triple 2286
\triplecovbond 2189
\trivlist 293
\tt 259
\ttbf 60
\ttdefault 561
\ttfamily 14, 16, 18, 53, 135, 160, 162, 164, 196, 197, 211, 231, 232, 234, 237, 266, 352, 358, 360, 361, 399, 401, 660, 733, 792, 872, 965, 1003, 1019, 1038, 1041, 1046, 1113, 1207, 1224, 1248, 1325, 1385, 1409, 1433, 1440, 1483, 1517, 1522, 1591, 1669, 1671, 1708, 1716, 1719, 1834, 1962, 1964, 1965, 1971, 1976, 1993, 1999, 2010, 2074, 2082, 2200, 2238, 2246, 2249, 2250, 2251, 2254, 2255, 2257, 2263, 2269, 2282, 2288, 2290, 2291, 2292, 2294, 2299, 2305, 2332, 2345, 2348, 2350, 2352, 2354, 2358, 2363, 2371, 2380, 2386, 2402, 2410, 2413, 2414, 2419, 2421, 2422, 2423, 2424, 2427, 2431, 2434, 2435, 2438, 2440, 2441, 2450, 2451, 2454, 2461, 2487, 2526, 2736, 2814, 2979, 3038, 3044, 3134, 3175
\tvcbox 2555, 2557, 2558
\two@digits 521
\twocolumn 445
\twocolumn[] 96
\twosymbolmessage 2572
\typein 37
typein 1
\typeout 34
\typesetcomment . 2887, 2900, 2980
typography rules
paragraph
indentation 594

U

\uline 1703
\Umd( ) a779
uncial (package) 201
\underbar 1
underline 772
Unicode 203
uninstall 1
\unitlength 328
\unitlength 1, 328, 332, 342
\UnitTested 806
\UnitTested 806, 1392
\unpenalty 2990, 3130
\unrestored@protected@xdef 51
\unsetattribute 542
\unskip 284
\uparrow 2228
\updownarrow 2229
\upshape 1031
\Upsilonmeson 2190
\url 2536
use commands:
\use:n . 315, 1335, 2319, 2346

\use_i:nnn 241
\use_ii:nnn 243
\use_none:n ... 357, 1699
\useasboundingbox 2771, 2775, 2787, 2791, 2801
\usebox 308
\usebox 2558
\usecounter 298
\usecounter 2548
\usefont 165
\usefont 214, 2097
\usefontcmdmessage 2607
\usepackage 515
\usepackage 493, 1926, 2030, 2047, 2049, 2099, 2216, 2322
\UseTextAccent 117
\usetextmathmessage 2597
\UseTextSymbol 117
\UseTextSymbol 109
\utfviii 783
\utfviii 2464
\utfviii 2465

V

\value 147
Values
asdf 727
auto 886
BAR 602
border-top-width 886
case 886
FOO 602
format 886
itemize 299
openbib 601
palette 886
pgf 763
\vardownarrow 2224
\varhookleftarrow 2225
\varhookrightarrow 2226
variable (environment) . 480, 805, 806
variable-sized symbols .. 781
\varleftarrow 2227
\varleftrightarrow 2228
\varlonghookrightarrow . 2229
\varlongleftarrow 2230
\varlongleftrightarrow . 2224
\varlongmapsfrom 2225
\varlongmapsto 2226
\varlongrightarrow 2227
\varmapsfrom 2228
\varmapsto 2229
\varnearrow 2230
\varnwarrow 2224
\varparsep 3074, 3076, 3082, 3086, 3092
\varphoton 2191
\varrightarrow 2225
\varsearrow 2226
\varswarrow 2227
\varuparrow 2228
\varupdownarrow 2229
\vbox 434, 2898
vbox commands:
\vbox_set 1228, 1315
\vbox_unpack_clear 1265, 1317
\vboxcolor 1837
vcoffin commands:
\vcoffin_set:Nnw 956
\vcoffin_set_end: ... 960
\vector 1
\verb 1
\Verbatim 407
\verbatim 259
\verbatim 406
\verbatimchar ... 2291, 2404, 2406, 2409, 2413, 2414, 2419, 2420
\verbatimfamily 20, 48, 311, 313
\verbatimfont . 16, 338, 393, 859, 940, 985, 3021, 3024, 3058, 3060, 3062, 3101, 3103
verse environment 580
version commands:
\l_phd_version 6
\vglue 22
\VIET 1928
\VIETfalse 2368
\Vl 2423
\Vp 782
\Vp 2421
\vphantom 264
\vphantom 188
\Vpl 2424
\vref 2600
\vskip 77, 84
\vspace 78, 86
\vspace 77

W

\water 2192
watermark (package) 425
\Wboson 2193

\Wbosonminus 2194
\Wbosonplus 2195
\wd 106, 2940
\Wf 2434
\whatsit 541
\wide accents 801
Wilson, Peter 207
\wlog 1, 2, 3, 4, 493, 2563, 2564, 2565
\write ... 94, 183, 188, 207, 423
\Wul 2436
\Ww 2435

X

\X 782
\x@protect 51
\xdef 197
XƎLaTeX 191
xindy 398
xkeyval (package) 486
xsect commands:
\xsect:n 94, 96, 100, 105, 108, 110, 159, 163, 172
\xspace 1748, 2465
\Xtemp 3147, 3154

Y

\Y 783
\year ... 147, 175, 219, 263, 330, 375, 418, 443, 502, 543, 589, 634, 678, 721, 765, 808, 851, 894, 1045, 1088, 1131, 1159, 1188, 1218, 1242, 1272, 1315, 1341, 1388, 1417, 1474, 1505, 1549, 1579, 1608, 1638, 1666, 1703, 1732, 1759, 1786, 1814, 1841, 1932, 1960, 1984, 2036, 2064, 2092, 2129, 2170, 2199, 2280, 2288, 2340
\Ytemp 3148, 3154
\yxdim 1

Z

\zap@space 516
\Zboson 2196

[12]

[12] 1.0,2013/01/26,Converted to DTX file.

phd-documentation Package, Code Implementation Objectives and Strategy

The package provides macros for documenting and indexing primarily LaTeX or LaTeX 3 code. It combines the styles originitating with `doc` with those found in the `pgf` series of packages. It removes some of the difficulties of using hyperlinks in the index and elsewhere in the document. 21.00002

16

-2.5

-2.1

0.87865

> "Lord Campbell [John Campbell, 1st Baron Campbell, 1779-1861] proposed that any author who published a book without an index should be deprived of the benefits of the Copyright Act; and the Hon. Horace Binney, LL.D. [1780-1875], a distinguished American lawyer, held the same views, and would have condemned the culprit to the same punishment."
>
> – H[enry] B[enjamin] Wheatley, How To Make An Index (New York: A. C. Armstrong & Son, 1902), 82

This package aims at providing authors with a set of tools and settings that can improve the typesetting of documentation and especially indices.

For the normal author there are both mark-up related macros, as well as a set of settings for indices.

My main motivation for developing this package was to group all the special documentation macros that I have used in developing the **phd** package. I also saw the need to hook up these settings with the concept of color palettes as described in the **phd-colorpalette** package. This enables the integration of full document templates.

C.1 Requirements Specification

1. To provide a declarative interface to enable users to modify documentation and index entries by setting keys, rather than writing macros. It is suggested that the author has a single style sheet that he loads to decorate a full document such as an article or a book or a document that describes code.
2. Setting of parameters will generally be with one command `\cxset`
3. Provide a consistent 'look and feel' for all code displays, irrespective of the package used, i.e., if **minted**, **listings** or **doc** is used the look and feel should be the same, as far as it is possible.
4. Provide a set of environments and commands for documenting symbols and fonts.
5. Seamless integration with doc type classes. As a minimal to support `ltxdoc` and **phddoc**. The latter is provided as part of the `phd` budle.
6. To provide a number of templates that cover most of the typical use case.
7. To provide an easy plug-in architecture for extensions.
8. Provide a detailed user manual.
9. Provide special indexing for doc

   `\stok`[⟨*number*⟩]{⟨*char*⟩}

   From egreg's italian latex guide `\stok` takes a `\<char>` argument and prints it with its category code or the number given as optional argument

Example: `\stok{a} or \stok{:}` a_{11} or $:_{12}$.

The development of this package predates **tcolorbox** and **minted**. As they are both excellent packages, I have abandoned some of my code and incorporated code from **tcolorbox** extensively.

C.2 Terminology

document Any written item, as a book, article, or letter, especially of a factual or informative nature.

heading A division of a document or document series. For a normal book headings are chapters, sections etc. However we allow for specifying a more complex document divided into books, volumes parts etc. For example the Bible has Books, chapters and verses, where a legal document might require divisions such as clauses. In general these divisions are numbered. These document divisions are stored in the comma list `phd_book_divisions_clist`.

head A typeset heading, such as chapter head, or section head. This can include a counter, label and title for example, *Chapter 1 Introduction*.

dom This is a programming interface that provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed. Although LaTeX does not offer a standard way to build such a tree (mainly because TeX does not require the marking of paragraphs, it is useful to think of the document as a tree structure. We also allow for a semi-automated way to build such a tree (with the exception that paragraphs are not included).

element A part of the document tree that can be styled on its own. For example the chapter label, or the section number.

C.3 Users

We classify users according to the LaTeX3 terminology as a) programmers b) template designers and c) authors.

C.3.1 Author

We assume that the author has an exising template which she is using but might want to do some minor modifications, for example use an italic shape for the font of the mark, but an upright font for the page numbers.

We follow the idea of representing the basic elements of documents as elements, each one having a parent in order to specify the element we need to style as accurate as possible. One can think of this approach being congruent with objects in other languages. As a matter fact nothing stops us from defining a key value interface as shown below.

```
\cxset
```

{
header.even.mark.font.size = `Large`,
header.even.mark.font.family = `serif`,
}

This would pehaps make it easier for the template designer, but I have rejected the idea as my aim is to make it easy for the author, who can search the template and just enter a couple of new proerty values.

C.3.2 Template designer

The template designer in the example above would have selected the format style from a number of predefined formats (templates) or would have created a style called *apa* from an existing template and modified it using declarative key style.

C.3.3 The programmer

The programmer in the example above could have created the basic format *apa* by using both declarative as well as defining or using existing macros. To the programmer we offer an extension mechanism, where the contents of a `ps@` command are defined. For example the programmer can define a new style using Ti*k*Z, but without having to worry about defining full `ps@` and their interface.

`\stok`[⟨*number*⟩]{⟨*char*⟩}

From egreg's italian latex guide `\stok` takes a `\<char>` argument and prints it with its category code or the number given as optional argument

Example: `\stok[a]` a_{11} or t_{12}.

C.4 Acknowledgements

This package couldn't have been possible if it was not for the documentation section of tcolorbox. I have liberally taken ideas and code from Dr. Thomas F. Sturm's package, which in turn draws strongly from the `PGF` manual. I am grateful to both.

C.5 Preliminaries

We declare that we use LaTeX and name the package. The code has been moved to the latest version of LaTeX to take advantage of more allocations. Package works well with LuaLaTeX. 7/8/2017

```
@@=
\wlog{*************************************************}
\wlog{ BEGIN PHD-DOCUMENTATION                          }
\wlog{*************************************************}
\NeedsTeXFormat{LaTeX2e}[2017/04/15]%
\ProvidesPackage{phd-documentation}[2017/04/15 v1.0 less preamble (YL)]%
```

`\MakePrivateLetters`

The **doc** provides the user with a command to define which characters can be treated as letters when scanning the contents of a `macrocode` environment. Since we are using l3 macros as well we redefine it as:

```
\DeclareDocumentCommand\MakePrivateLetters{}%
  {\catcode`\@11\relax \catcode`_11 \catcode`:11}
\ExplSyntaxOn
\definecolor{thecodelinecolor}{HTML}{140F15}
%\definecolor{thecodelinecolor}{HTML}{420943}
\tl_replace_once:Nnn {\theCodelineNo}
  {\HDorg@theCodelineNo}
  {{\color{thecodelinecolor}\ttfamily\scriptsize\arabic{CodelineNo}}}
\ExplSyntaxOff
```

We also need the **refcount**.

C.6 tcolorbox

We load **tcolorbox** with options theorems, skins, documentation etc for internal and external use.

We also provide an interface, between the **tcolorbox** documentation keys and our own.

The indexing keys are still to be sorted out with other sections of the documentation, but they seem to be working for the moment.

C.7 Verbatim fonts

There are different requirements for verbatim fonts. The module **phd-fontmanager** is our main module for defining fonts. The main mono font is set using the key `/phd/main mono font-face`. This is stored in `\main_mono_fontface_tl`. In this document it is set as: macro:->Source Code Pro

```
\newcommand*{\verbatimfont}{\ttfamily}%
\let\displayverbfont\ttfamily
% defined in verbatim package
\renewcommand*{\verbatim@font}{\verbatimfont}
```

The **tcolorbox** loads the packages **pgf**, **verbatim**, **environ**, **etoolbox** when the package is loaded. It also loads more when other libraries are loaded.

```
% Loaded after in pkg manager
% Someone is loading marvosym here and this gives problems
\let\oldherm\Hermaphrodite
\let\Hermaphrodite\undefined

\RequirePackage{tcolorbox}[2018/07/26] %version 4.14 latest
\tcbuselibrary{documentation,listings,skins,xparse,external,breakable,theorems}
\tcbset{index format=doc,
        index actual={=},
        index level = {>},
        index quote = {!},
```

```
        index german settings,
        color hyperlink = thelinkcolor ,
        color definition =thelinkcolor,
        index command= \index,
                    }
\tcbset{halostyle/.style={fuzzy halo=2mm with magenta!5}}
```

We use the **docCommand** as `\DescribeMacro` when in doc, hence we better mark it as a `\SpecialMainIndex`.

phdd_label_prefix_tl

```
\ExplSyntaxOn
\def\l_phdd_label_prefix_tl {com:}
```

(*End definition for* `\l_phdd_label_prefix_tl`.)

```
\renewenvironment{docCommand}[3][]{\tcbset{#1}%
  \begin{tcb@manual@entry}%
  \begin{tcb@doc@head}{doc@head@command}%
  % Print the command using the current style
  \tcb@Print@Com{#2}
  %{\verbatimfamily\bfseries\textbackslash\color{themacrocolor}\tl_to_str:n{#2}}
  \SpecialMainIndex{#2}%
  \protected@edef\@currentlabel{\noexpand\tcb@cs{#2}}\label{com:#2}{\ttfamily #3}%
  \tcb@doc@do@description%
  \end{tcb@doc@head}\nobreak\tcbset{before~upper=}\kvtcb@doc@body@command@before\ignores
  {\ifvmode\else\unskip\fi\kvtcb@doc@body@command@after\end{tcb@manual@entry}}


%\renewenvironment{docCommand*}{\bgroup\tcb@doc@toindexfalse\begin{docCommand}}{\end{doc
\ExplSyntaxOff
%\def\tcb@Print@Com#1{\textcolor{\kvtcb@col@command}{\ttbf\char`\\ \detokenize{#1}}}

```

C.8 Keys dealing with colors

The actual color definitions should come from a palette, using **phd-colorpalette**. They have names such as `themacrocolor`.

The color names are stored in token lists. The module colorizer, colors code in a macrocode environment and has similar keys.

```
\ExplSyntaxOn
\cxset{
    color~command/.store~in        = \phdd_key_color_command_tl,
    color~environment/.store~in    = \phdd_key_color_environment_tl,
    color~key/.store~in            = \phdd_key_color_tl,
    color~value/.store~in          = \phdd_key_color_value_tl,
    color~color/.store~in          = \phdd_key_color_color_tl,
    color~definition/.style        = {color~command={#1},
                                      color~environment={#1},
                                      color~key={#1},
                                      color~value={#1},
                                      color~color={#1}},
    color~option/.store~in         = \phdkv@col@opt,
    color~hyperlink/.store~in      = \phdkv@colhyper,
```

```
    color~frame/.store~in          = \phdkv@colhyper,
    color~meta/.store~in           = \pkdkv@colmeta,
%
    before~example/.store~in       = \phdkv@beforeexample,
    after~example/.store~in        = \phdkv@afterexample,
}
\ExplSyntaxOff
```

```
/DOCUM
*DOCUM|DFLT
DOCUM\cxset {
   color command     = themacrocolor,
   color environment = theenvironment,
   color key         = thekey,
   color value       = thevalue,
   color color       = black,%leaks to index
   color option      = theoption,
   color meta        = themeta,
   color frame       = theframe,
DOCUM }
/DOCUM|DFLT
*DOCUM
```

\docAuxCommand{⟨*command name*⟩}

The \docAuxCommand is used to typeset and index an inline command. The star version will just typeset the command. If we are using the **phddoc** or the **ltxdoc** we have to add the `encapchar hyperpage` otherwise the links do not appear in the index.

d_index_typeset_bool A boolean to indicate if a variable is to be indexed or not. Used for commands.

```
\ExplSyntaxOn
\bool_new:N \phdd_index_typeset_bool
\bool_gset_true:N \phdd_typeset_bool
```

(*End definition for* `\phdd_index_typeset_bool`.)

```
\cs_new:Npn \phdd_typeset_command #1
  {
    \group_begin:
    \color{\phdd_key_color_command_tl}
    \cs[no-index]{#1}
    \group_end:
  }

\cs_set:Npn \phdd_typeset_index_command #1
  {
    \bool_if:NT \phdd_index_typeset_bool
      {
        \kvtcb@index@command
          {
            #1\idx@actual{\tcbIndexPrintComC{#1}}
            \encapchar hyperpage
          }
      }
```

```
    \phdd_typeset_command {#1}
  }

%\DeclareDocumentCommand\docAuxCommand { s m }
%  {
%    \phdd_typeset_index_command {#2}
%    \IfBooleanTF{#1}{}{\tcb@index@Com{#2}}
%  }
%\ExplSyntaxOff
```

\docColor{⟨*color name*⟩}

Typesets a color name and also adds it onto the index. This is identical with tcolorbox, which we overwrite.

Example: The thecs is the color used for commands.

\phdindexprintcolc
\phdd_print_col
phdd_doc_color_aux:n
hdd_doc_color_star:n

```
\ExplSyntaxOn
\newrobustcmd{\phdindexprintcolc}[1]
  {
    \phd_index_print_c{#1}{\phdd_key_color_color_tl}
  }

\cs_set:Npn \phd_print_col #1
  {
    \textcolor{\phdd_key_color_color_tl}{\ttfamily\bfseries#1}
  }

\cs_set:Npn \@@_doc_color_aux:n #1
  {
    \phd_print_col {#1}
    \phd_index_col {#1}
  }

\cs_set:Npn \@@_doc_color_star:n #1
  {
    \phd_print_col{#1}
  }

\DeclareDocumentCommand {\docColor} { s m }
  {
    \IfBooleanTF {#1}
      { \@@_doc_color_star:n {#2} }
      { \@@_doc_color_aux:n {#2} }
  }
\ExplSyntaxOff
```

(*End definition for* \phdindexprintcolc *and others.*)

\docValue{⟨*value*⟩}

Typesets a value asdf.

phd_index_val
\docValue@
\docValue@star

```
\ExplSyntaxOn
\cs_set:Npn \phd_index_val #1
  {
    \ifphd@doc@toindex
      \index
      {
        #1\idx@actual
        \phdindexprintvalca{#1}
      }
      \index
        {
          \phdkv@text@values
          \idx@level #1
          \idx@actual
          \phdIndexPrintValC{#1}
        }
    \fi
  }


\cs_set:Npn \docValue@ #1 {\phd@Print@Val{#1}\phd_index_val{#1}}%
\cs_set:Npn \docValue@star #1 {\phd@Print@Val{#1}}%

\DeclareDocumentCommand \docValue { s m }
  {
    \IfBooleanTF {#1}
      { \docValue@star {#2} }
      { \docValue@ {#2} }
  }
\ExplSyntaxOff
```

(*End definition for* `phd_index_val`, `\docValue@`, *and* `\docValue@star`.)

```
\cxset {doc command color/.code = \tcbset{color command = #1}}
\cxset {doc command color= thecmdcolor}
%
%
\lstdefinelanguage{extras}{morekeywords={%
      poemtitle, poemtoc, versewidth,
      vin, poemlines,poemtitlefont,
      ProvidesClass,IfFileExists,
      RequirePackage,ifthenelse,chapter,
      includegraphics, newarray,readarray,of
}}
\lstloadlanguages{[LaTeX]TeX, [primitive]TeX, extras}
```

Note the **gobble=1** option. We use this to make the colorboxes with code not to show the '%' sign in this documentation. Ideally you should fork the code below and adapt it to your own needs.

Also note that this is the default that is to be used in **tcolorbox** commands.

`\begin{scriptexample}`[⟨*options*⟩]
⟨*text*⟩
`\end{scriptexample}`

Display box for examples of various languages and scripts.

```
    \newtcolorbox{scriptexample}[2][]{colback=thecodebackground,
    boxrule=0pt,toprule=0pt,colframe=white,#1}
```

```
\newtcolorbox{commands}[2][]{colback=thecodebackground,
boxrule=0pt,toprule=0pt,colframe=white}
```

Set the **listings** defaults for TeX and LaTeX. All colours are harmonized with the definitions od the color palette, as set in the **phd-colorpalette**.

```
\lstset{language={[LaTeX]TeX},
       escapeinside={{(*@}{@*)}},
       numbers=left,
       gobble=0,
       stepnumber=1,
       numbersep=5pt,
       numberstyle={\footnotesize\color{thegray}},
       breaklines=false,
       framesep=5pt,
       basicstyle=\small\ttfamily,
       showstringspaces=false,
       stringstyle={\color{orange}\footnotesize},
       commentstyle=\color{black},
       rulecolor=\color{theshade},
       breakatwhitespace=true,
       showspaces=false,
       xleftmargin=10pt,
       xrightmargin=0pt,
       aboveskip=3pt plus1pt minus1pt,
       belowskip=7pt plus1pt minus1pt,
       backgroundcolor=\color{theshade},
}
```

We predefine some listings styles to use with color palettes. First we define three styles for TeX and LaTeX, as well as three environements.`simple` [1]

```
\def\footnotechanges#1#2#3{\footnote{#1#2#3}}
\lstdefinestyle{simple}{%
       escapeinside={{(*@}{@*)}},
       numbers=left,
       gobble=0,
       stepnumber=1,
       numbersep=5pt,
       firstnumber=last,
       numberstyle={\ttfamily\scriptsize\color{thecodelinecolor}},
       breaklines=true,
       % we do not use frames
       framesep=0pt,
       framexleftmargin=0pt,
       basicstyle=\small\ttfamily,
       showstringspaces=true,
       stringstyle={\color{thestringstyle}\footnotesize},
       commentstyle=\color{thecommentstyle},
       rulecolor=\color{theshade},
       breakatwhitespace=true,
       prebreak={\Righttorque},
```

[1] v1.01 2019/01/26 Made lines to line up with codeline.

```
        postbreak={\space\Lefttorque},
        showspaces=false,
        xleftmargin=1.5pt,
        xrightmargin=0pt,
        aboveskip=3pt plus1pt minus1pt,
        belowskip=3pt plus1pt minus1pt,
        backgroundcolor=\color{white},
        showlines=false,
        gobble=0,
        resetmargins,
}

\lstdefinestyle{singleline}{%
        escapeinside={{(*@}{@*)}},
        numbers=left,
        gobble=0,
        stepnumber=1,
        numbersep=5pt,
        numberstyle={\footnotesize\color{thecodelinecolor}},
        breaklines=false,
        framesep=0pt,
        framexleftmargin=0pt,
        basicstyle=\small\ttfamily,
        showstringspaces=false,
        stringstyle={\color{thestringstyle}\footnotesize},
        commentstyle=\color{thecommentstyle},
        rulecolor=\color{theshade},
        breakatwhitespace=true,
        prebreak={\Righttorque},
        postbreak={\space\Lefttorque},
        showspaces=false,
        xleftmargin=-10pt,
        xrightmargin=0pt,
        aboveskip=3pt plus1pt minus1pt,
        belowskip=3pt plus1pt minus1pt,
        backgroundcolor=\color{white},
        showlines=false,
        numbers=none,
        resetmargins,
}


% prints the style
\def\phd@print@LstStyle#1{%
  \textcolor{\kvtcb@col@counter}{\tcb@doc@bfseries\tcb@scantokens{#1}}
}

\DeclareDocumentCommand\docAuxListingsStyle{s m}{%
  \phd@print@LstStyle{#2}%
  \IfBooleanTF{#1}{}{%
  %\index{\phd@text@lststyles\idx@level#2}
  \SpecialPackageIndex[\phd@text@lststyles]{#2}
  }%
}
```

An example of `\docAuxListingsStyle` `simplex` .

The environment `\begin{TeX}..\end{TeX}` provides a listings environment for typesetting, either TeX or LaTeX code. This environment always starts the number

```
\lstnewenvironment{teX}[1][]
  {\lstset{language=[LaTeX]TeX}\lstset{%
      style=simple,#1
}}
{}
```

`\begin{sverbatim}`[⟨*options*⟩]
⟨*environment content*⟩
`\end{sverbatim}`

A listings environment for single line verbatim typesetting. It uses the single line listings style.`singleline` . It typesets with smaller values for `aboveskip` and `belowskip`.

```
\lstnewenvironment{sverbatim}[1][]
  {\lstset{language=[LaTeX]TeX}\lstset{%
      style=singleline,#1
}}
{}
```

The `extended` is similar to the rest, except it is numbered continuouly.

```
\lstdefinestyle{extended}{%
      breaklines=true,
      framesep=5pt,
      basicstyle=\verbatimfamily,
      showstringspaces=false,
      keywordstyle=\small\verbatimfamily,
      stringstyle=\color{thestringstyle},
      commentstyle=\color{thecommentstyle},
      rulecolor=\color{gray!10},
      breakatwhitespace=true,
      xleftmargin=10pt,
      xrightmargin=0pt,
      aboveskip=\medskipamount,
      belowskip=\medskipamount,
      backgroundcolor=\color{thecodebackground},
}
```

`\begin{teXX}`[⟨*listings options*⟩]
⟨*code*⟩
`\end{teXX}`

Different listings environment.

```
\lstnewenvironment{teXX}[1][]
  {\lstset{language=[LaTeX]TeX}\lstset{%
      style=extended, #1
}}
{}
```

`\continuelinenumber`

Continues code numbers from previous block

```
\newcommand\continuelinenumber{\lstset{firstnumber=last}}
```

startnumberat The macro **\continueLineNumber**, provides a command to start the next block of code with the code numbers continuing. This requires the **listings** which is already included.

\startlineat⟨*line number*⟩

Starts a listings display with ⟨*line number*⟩.

```
% Always I forget this so I created some aliases
\DeclareDocumentCommand{\startlineat}{ m }{\lstset{firstnumber=#1}}
```

\numberlineat⟨*line number*⟩

An alias for **\startlineat**.

```
\let\numberlineat\startlineat
\let\startnumberat\numberlineat
```

```
\newcommand\emphasis[2][black!80]{%
   \lstset{%
     emph={write, writeln,#2},
     escapeinside={(*@}{@*)},
     emphstyle={\verbatimfont%
                \bfseries%
                \textcolor{#1}%
                },
   }%
}
```

\begin{teXXX}[⟨*options*⟩]
⟨*environment content*⟩
\end{teXXX}

Environment for typestting long tracks of code with full highlighting of keywords, uses the **highlite** style.

```
\lstdefinestyle{highlite}{%
      alsolanguage=makeindex,
      alsolanguage=BibTeX,
      emph={cs, use,new,seq,map,inline,eq,gincr,incr,IfNoValueF,%
      if,If,exist,protect,nopar,gset,%
      set,undefine,define,add,gadd,remove,div,newcounter%
      round,truncate,max,min,mod,gzero,int,%
      zero,newcount,protected,msg,error,DeclareDocumentCommand},
      emphstyle=\ttfamily\color{thered},
      firstnumber=last,
      stepnumber=1,
      escapeinside={{(*@}{@*)}},
      breaklines=true,
      framesep=5pt,
      basicstyle= \ttfamily,
      showstringspaces=false,
      keywordstyle=\ttfamily\color{thered},%\color{primary},
      keywordstyle=[2]\ttfamily\color{black},
```

```
      stringstyle=\color{thecommentscolor},
      commentstyle=\color{thecommentscolor},
      rulecolor=\color{gray!10},
      breakatwhitespace=true,
      prebreak={\Righttorque},
      postbreak={\space\Lefttorque},
      showspaces=false,  % shows spacing symbol
      upquote=true,
      xleftmargin=10pt,
      xrightmargin=0pt,
      aboveskip=\medskipamount,
      belowskip=\medskipamount,
      backgroundcolor=,
      tabsize=2,%
      mathescape=true,%
}

\lstnewenvironment{teXXX}[1][]
  {\lstset{language=[LaTeX]TeX}%
    \lstset{%
      style=highlite,  #1
}}
{}
\lstnewenvironment{phdverbatim}[1][]
  {\lstset{language=[LaTeX]TeX}%
    \lstset{%
      emph={cs, use,new,seq,map,inline,eq,gincr,incr,IfNoValueF,%
      if,If,exist,protect,nopar,gset,%
      set,undefine,define,add,gadd,remove,div,%
      round,truncate,max,min,mod,gzero,int,%
      zero,newcount,protected,msg,error,DeclareDocumentCommand},
      emphstyle=\verbatimfont\bfseries\color{black!80},
      numbers=none,
     % stepnumber=1,
      escapeinside={{(*@}{@*)}},
      breaklines=false,
      framesep=5pt,
      basicstyle= {\small\ttfamily},
      showstringspaces=false,
      keywordstyle=\ttfamily\color{thekeywordstyle},
      stringstyle=\color{black!50},
      commentstyle=\color{black!50},
rulecolor=\color{gray!10},
      breakatwhitespace=true,
      showspaces=false,  % shows spacing symbol
xleftmargin=15pt,
xrightmargin=5pt,
%  aboveskip=0pt, % compact the code looks ugly in type
% belowskip=0pt,  % user responsible to insert any skips
aboveskip=\medskipamount,
belowskip=\medskipamount,
       backgroundcolor=,
       #1
}}
```

```
{}
```

The following have been taken from `https://github.com/cgnieder/cnltx/blob/master/cnltx-listings.sty`

```
\lstdefinelanguage{makeindex}{
  morekeywords = {
    actual,           % @
    arg_open,         % {
    arg_close,        % }
    encap,            % |
    escape,           % \\
    keyword,          % \\indexentry
    level,            % !
    page_compositor,  % -
    quote,            % "
    range_open,       % (
    range_close,      % )
    preamble,         % \\begin{theindex} \n
    postamble,        % \n\n \end{theindex} \n
    setpage_prefix,   % \n \\setcounter{page}{
    setpage_suffix,   % } \n
    group_skip,       % \n\n \\indexspace \n
    headings_flag,    % 0
    heading_prefix,   %
    heading_suffix,   %
    symhead_positive, % Symbols
    symhead_negative, % symbols
    numhead_positive, % Numbers
    numhead_negative, % numbers
    item_0,           % \n \\item
    item_1,           % \n \\subitem
    item_2,           % \n \\subsubitem
    item_01,          % \n \\subitem
    item_x1,          % \n \\subitem
    item_12,          % \n \\subsubitem
    item_x2,          % \n \\subsubitem
    delim_0,          % ,
    delim_1,          % ,
    delim_2,          % ,
    delim_n,          % ,
    delim_r,          % --
    delim_t,          %
    suffix_2p,        %
    suffix_3p,        %
    suffix_mp,        %
    encap_prefix,     % \\
    encap_infix,      % {
    encap-suffix,     % }
    line_max,         % 72
    indent_space,     % \t\t
    indent_length     % 16
  } ,
  morestring = [b]{"} ,
  morecomment = [l]{\%} ,
```

```
  sensitive = true
}
% ---------------------------------------------------------------------------
% a listings language BibTeX:
\lstdefinelanguage{BibTeX}{
  % entry types:
  morekeywords = {
    % regular types:
    @article,
    @book,@mvbook,@inbook,@bookinbook,@suppbook,@booklet,
    @collection,@mvcollection,@incollection,@suppcollection,
    @manual,
    @misc,
    @online,
    @patent,
    @periodical,
    @suppperiodical,
    @proceedings,@mvproceedings,@inproceedings,
    @reference,@mvreference,@inreference,
    @report,
    @set,
    @thesis,
    @unpublished,
    @xdata,
    @customa,@customb,@customc,@customd,@custome,@customf,
    % type aliases:
    @conference,
    @electronic,
    @mastersthesis,
    @phdthesis,
    @techreport,
    @www,
    % unsupported types:
    @artwork,
    @audiobibnote,
    @commentary,
    @image,
    @jurisdiction,
    @legislation,
    @legal,
    @letter,
    @movie,
    @music
    @performance,
    @review,
    @software,
    @standard,
    @video,
    % cnltx types:
    @bundle,
    @class,
    @package,
  } ,
  % entry fields:
```

```
morekeywords = [2]{
  % data fields:
  abstract,
  addendum,
  afterword,
  annotation,
  annotator,
  author,
  authortype,
  bookauthor,
  bookpagination,
  booksubtitle,
  booktitle,
  booktitleaddon,
  chapter,
  commentator,
  date,
  doi,
  edition,
  editor,editora,editorb,editorc,
  editortype,
  editoratype,editorbtype,editorctype,
  eid,
  entrysubtype,
  eprint,
  eprintclass,
  eprinttype,
  eventdate,
  eventtitle,
  eventtitleaddon,
  file,
  foreword,
  hvarer,
  howpublished,
  indextitle,
  institution,
  introduction,
  isan,
  isbn,
  ismn,
  isrn,
  issn,
  issue,
  issuesubtitle,
  issuetitle,
  iswc,
  journalsubtitle,
  journaltitle,
  label,
  language,
  library,
  location,
  mainsubtitle,
  maintitle,
```

```
    maintitleaddon,
    month,
    nameaddon,
    note,
    number,
    organization,
    origdate,
    origlanguage,
    origlocation,
    origpublisher,
    origtitle,
    pages,
    pagetotal,
    pagination,
    part,
    publisher,
    pubstate,
    reprinttitle,
    series,
    shortauthor,
    shorteditor,
    shorthand,
    shorthandintro,
    shortjournal,
    shortseries,
    shorttitle,
    subtitle,
    title,
    titleaddon,
    translator,
    type,
    url,
    urldate,
    venue,
    version,
    volume,
    volumes,
    year,
    % special fields:
    crossref,
    entryset,
    execute,
    gender,
    hyphenation,
    ids,
    indexsorttitle,
    keywords,
    options,
    presort,
    related,
    relatedoptions,
    relatedtype,
    relatedstring,
    sortkey,
```

```
    sortname,
    sortshorthand,
    sorttitle,
    sortyear,
    xdata,
    xref,
    % custom fields:
    namea,nameb,namec,
    nameatype,namebtype,namectype,
    lista,listb,listc,listd,liste,listf,
    usera,userb,userc,userd,usere,userf,
    verba,verbb,verbc,
    % field aliases:
    address,
    annote,
    archiveprefix,
    journal,
    key,
    pdf,
    primaryclass,
    school,
    % cnltx fields:
    maintainer
  } ,
  morestring = [b]{"} ,
  morecomment = [l]{\%} ,
  sensitive = false
}

% listings style for source code from igniter
\lstdefinestyle{complex}{
  basicstyle       =  \small\ttfamily,  %{\sourceformat},
  numbers          = left,
  numberstyle      = \tiny,
  xleftmargin      = 1em,
  numbersep        = .75em,
  gobble           = \@gobble ,
  columns          = fullflexible,
  literate         =
    {ä}{{\"a}}{1}
    {ö}{{\"o}}{1}
    {ü}{{\"u}}{1}
    {Ä}{{\"A}}{1}
    {Ö}{{\"O}}{1}
    {Ü}{{\"U}}{1}
    {ß}{{\ss}}{1}
    {`}{{\`{}}}{1}
    {'}{\textquotesingle}{1} ,
  breaklines       = true,
  keepspaces       = true,
  breakindent      = 1em,
  commentstyle     = \color{thecomment},
  keywordstyle     = \color{thecs},
  deletetexcs      =
```

```
    {
      a,o,u,A,O,U,
      begin,
      center,
      description,document,
      end,enumerate,
      equation,eqnarray,
      figure,flushleft,flushright,
      itemize,list,
      otherlanguage,
      table,tabu,tabular
    },
  deletekeywords   =
    {
      a,o,u,A,O,U,
      begin,
      center,
      description,document,
      end,enumerate,
      equation,eqnarray,
      figure,flushleft,flushright,
      itemize,list,
      otherlanguage,
      table,tabu,tabular
    },
  % \begin, \end:
  texcsstyle       = [2]\color{red}, %add color begin end
  index            = [2][texcs2],
  indexstyle       = [2]\@gobble,
  moretexcs        = [2]{begin,end},
  % control sequences that'll be indexed:
  texcsstyle       = [3]\color{cs},
  index              = [3][texcs3],
  indexstyle       = [3]\indexcs,
  % control sequences that won't be indexed:
  texcsstyle       = [4]\color{thecs},
  index            = [4][texcs4],
  indexstyle       = [4]\@gobble,
  % added environments that'll be indexed:
  texcsstyle       = [5]\color{env},
  index            = [5][texcs5],
  %indexstyle       = [5]\indexenv,   UNCOMMENT LATER
  % environments that won't be indexed:
  texcsstyle       = [6]\color{env},
  index            = [6][texcs6],
  indexstyle       = [6]\@gobble,
  moredelim        = *[s][\color{green}]{$}{$} %add color the math
}
\lstnewenvironment{lualisting}[1][]
{\lstset{language=[LaTeX]TeX,
  basicstyle           = \ttfamily,
  showstringspaces     = false,
  upquote              = true,
  keywordstyle         =\color{blue},
```

```
  commentstyle          =\color{black!50},
  stringstyle           =\color{black!80},
  backgroundcolor       =\color{white},
  xleftmargin           =15pt,
  xrightmargin          =5pt,
  aboveskip             =\medskipamount,
  belowskip             =\medskipamount,
  #1}}
{}
```

C.8.1 Go language

```
\lstdefinelanguage{Go}{
  % Keywords as defined in the BNF
    morekeywords=[1]{break,default,func,interface,%
      case,defer,go,map,struct,chan,else,goto,package,%
      switch,const,fallthrough,if,range,type,continue,%
      for,import,return,var,select},
    % Special identifiers, builtin functions
    morekeywords=[2]{make,new,nil,len,cap,copy,complex,%
        real,imag,panic,recover,print,println,iota,close,%
        closed,_,true,false,append,delete},
    % Basic types
    morekeywords=[3]{%
        string,int,uint,uintptr,double,float,byte,%
          int8,int16,int32,int64,int128,%
          uint8,uint16,uint32,uint64,uint128,%
          float32,float64,complex64,complex128,%
          rune},
    % Strings : "toto", 'toto', `toto`
    morestring=[b]{"},
    morestring=[b]{'},
    morestring=[b]{`},
    % Comments : /* comment */ and // comment
    comment=[l]{//},
    morecomment=[s]{/*}{*/},
    % Options
    sensitive=true,
    numbers=left,
    numberstyle=\footnotesize
}
\lstnewenvironment{Go}[1][]
  {\lstset{language=Go}%
    \lstset{%
      style=highlite,  #1
}}
{}
```

```
package main

import (
→"flag"
```

```
→"os"
→"github.com/involucro/involucro/app"  ❶
→"github.com/involucro/involucro/ilog"
 )

 func main() {
→err := app.Main(os.Args)

→switch err {
→case flag.ErrHelp:
→→os.Exit(0)
→case nil:
→→os.Exit(0)
→default:
→→ilog.Error.Logf("Task processing failed: %s", err)
→→os.Exit(1)
→}
 }
```

See ❶

C.9 LaTeX code demo environments

To demonstrate LaTeX code it is sometimes desirable to have the code be executed. This was pioneered in a number of packages. One of the better packages to do so is **tcolorbox**. We use it to define a special environment.

`\begin{texexample}` {⟨*title*⟩} {⟨*label for referencing*⟩}
⟨*environment content*⟩
`\end{texexample}`

The environment `texexample` will list the code using the **listings** package, so we can have a nice box and shows the output at the bottom section.

First we define a new counter which resets at every chapter. If `c@chapter` is not defined we reset it based on sections.

#1 Title of the example

#2 label for referencing

```
\ifx\c@chapter\@undefined
  \newcounter{texexp}[section]
  \@addtoreset{c@texexp}{c@section}
\else
  \newcounter{texexp}[chapter]
  \@addtoreset{c@texexp}{c@chapter}
\fi
```

```
%\tcbset{listing utf8=latin1}% optional; 'latin1' is the default.
\def\dcircle#1{\ding{\numexpr181 + #1\relax}}
\def\thetexexp{\@arabic\c@section.\arabic{texexp}}
```

```
\tcbset{texexp/.style={%
    fonttitle=\small\ttfamily,
```

```
    fontupper=\small,
    fontlower=\small,
    coltitle=black,
    colback = thecodebackground,% background
    colframe=thecodebackground,
  % process code={\def\dcircle##1{\ding{\numexpr181 + ##1}}},
      %colupper=spot!,
  },
  listing options = {%
    gobble=2
    keywordstyle=\color{thekeywordstyle},
    belowskip=0pt,
    escapeinside={(*@}{@*)},%
    breaklines=true,%
    backgroundcolor=\color{thecodebackground},%
    %firstnumber=last,%
    stepnumber=1,%
    upquote=true,%
    alsoletter={_,:},%
    commentstyle=\color{thecommentscolor},%
    emph={cs,new,seq,map,inline,eq,gincr,incr,IfNoValueF,if,%
           If,exist,protect,nopar,gset,%
           set,undefine,define,add,gadd,remove,div,%
           round,truncate,max,min,mod,gzero,int,exp,put,left,args,%
           zero,newcount,protected,msg,error,%
           eval,to,arabic,alph,Alph,roman,Roman,dim%
           DeclareDocumentCommand,%
           NewDocumentCommand,%
           RenewDocumentCommand,includegraphics,
           function,local,return,break,
         },%
           %
          % For LaTeX3 we need to add these, note % is important
          % dn't miss, at the end...
          moretexcs    = {DeclareDocumentCommand,IfBooleanTF,tex_def:D,%
          cs_new:Nn,cs_new:Npn,cs_new:cn,cs_set_nopar:Npn,cs_meaning:N,%
          %primitives
          cs:w,cs_end:,tex_underline,group_begin:, group_end:,%
          %coffins
          NewCoffin,JoinCoffins,SetHorizontalCoffin,TypesetCoffin,%
          %properties
          prop_new:N,prop_new:c,prop_put:Nnn,%
          %boolean
          bool_new:N,bool_set_true:N,bool_set_false:N,%
          bool_if:NTF,%
          hbox_to_wd:nn,%
          IfNoValueTF,%
          %token lists
          tl_new:N,tl_set:Nn,tl_concat:NNN,%
          token_to_meaning:N,%
          seq_pop_left:NN,%
          %
          %int
          int_if_exist:cT,int_use:c,int_new:c,int_new:N,int_eval:n,%
```

```
          int_add,int_use,int_to_roman,%
          %boxes
          box_new:c,hbox_set:cn,box_use:c,vbox_set:cn,box_move_down:nn,%
          %string
          str_if_eq_x:nnTF,%
          tl_tail:n,%
          DeclareObjectType,%
          DeclareTemplateInterface,%
          DeclareTemplateCode,%
          DeclareInstance,UseInstance,AssignTemplateKeys%
          keys_set,keys_define,%
          },%
         emphstyle=\verbatimfont\bfseries\color{theemphasiscolor},%
         %framexleftmargin=-14pt,numbersep=-7pt,xleftmargin=-28pt,
   },%close listings options
      % added for better control
      arc=0pt,
      outer arc=0pt,
      example1/.code 2 args={\refstepcounter{texexp}{\ifx#2\empty\else\label{#2}\fi}}%Re
     \pgfkeysalso{texexp, enhanced, breakable, title={Example \thetexexp\ #1}%
 },
}

\def\emphasize#1{%
\tcbset{texexp/.style={%
    fonttitle=\small\ttfamily,
    fontupper=\small,
    fontlower=\small,
    coltitle=black,
    colback = thecodebackground,% background
    colframe=thecodebackground,
      %process code={\def\dcircle##1{\ding{\numexpr181 + ##1}}},
      %colupper=spot!,
   },
   listing options = {%
     gobble=1,
     tabsize=2,
     keywordstyle=\color{thekeywordstyle},
     belowskip=0pt,
     escapeinside={(*@}{@*)},%
     breaklines=true,%
     backgroundcolor=\color{thecodebackground},%
     %firstnumber=last,%
     stepnumber=1,%
     upquote=true,%
     alsoletter={_,:},%
     commentstyle=\color{thecommentstyle},%
     emph={cs,new,seq,map,inline,eq,gincr,incr,IfNoValueF,if,%
            If,exist,protect,nopar,gset,%
            set,undefine,define,add,gadd,remove,div,%
            round,truncate,max,min,mod,gzero,int,exp,put,left,args,%
            zero,newcount,protected,msg,error,%
            eval,to,arabic,alph,Alph,roman,Roman,dim%
```

```
            DeclareDocumentCommand,%
            NewDocumentCommand,%
            RenewDocumentCommand,includegraphics,
            function,local,return,#1,
         },%
           %
          % For LaTeX3 we need to add these, note % is important
          % dn't miss, at the end...
     moretexcs    = {DeclareDocumentCommand,IfBooleanTF,tex_def:D,%
          cs_new:Nn,cs_new:Npn,cs_new:cn,cs_set_nopar:Npn,token_to_meaning:N,%
          %primitives
          cs:w,cs_end:,tex_underline,group_begin:, group_end:,%
          %coffins
          NewCoffin,JoinCoffins,SetHorizontalCoffin,TypesetCoffin,%
          %properties
          prop_new:N,prop_new:c,prop_put:Nnn,%
          %boolean
          bool_new:N,bool_set_true:N,bool_set_false:N,%
          bool_if:NTF,%
          hbox_to_wd:nn,%
          IfNoValueTF,%
          %token lists
          tl_new:N,tl_set:Nn,tl_concat:NNN,%
          token_to_meaning:N,%
          seq_pop_left:NN,%
          %
          %int
          int_if_exist:cT,int_use:c,int_new:c,int_new:N,int_eval:n,%
          int_add,int_use,int_to_roman,%
          %boxes
          box_new:c,hbox_set:cn,box_use:c,vbox_set:cn,box_move_down:nn,%
          %string
          str_if_eq_x:nnTF,%
          tl_tail:n,%
          DeclareObjectType,%
          DeclareTemplateInterface,%
          DeclareTemplateCode,%
          DeclareInstance,UseInstance,AssignTemplateKeys,%
          keys_set,keys_define,%
          },%
     emphstyle=\verbatimfont\bfseries\color{theemphasiscolor},%
    %framexleftmargin=-14pt,numbersep=-7pt,xleftmargin=-28pt,
    %
   },%close listings options
      % added for better control
      arc=0pt,
      outer arc=0pt,
  }%close style
}%close command
%
\newenvironment{texexp}[1]{\tcblisting{texexp,#1}}{\endtcblisting}

\newenvironment{example1}[3][]{\tcblisting{example1={#2}{#3},#1}}%
    {\endtcblisting}
```

```
%
```

\begin{texexample} [⟨⟩] {⟨*Title*⟩} ⟨*label*⟩
⟨*environment content*⟩
\end{texexample}

```
\newenvironment{texexample}[3][]{\noindent\tcblisting{example1={#2}{#3},#1}}%
    {\endtcblisting }

\newenvironment{texcode}[3][listing only]{\noindent\tcblisting{example1={#2}{#3},#1}}%
    {\endtcblisting }
%
% Need to fix
\let\luaexample\texexample
\let\endluaexample\endtexexample
%\tcbset{luacode/.style={%
%      fonttitle=\small\ttfamily,
%      fontupper=\small,
%      fontlower=\small,
%      coltitle=black,
%      colback = thecodebackground,% background
%      colframe=thecodebackground,
%      %colupper=spot!,
%      },
%      listing options = {
%          language={[5.2]Lua},
%          belowskip=0pt,
%          escapeinside={(*@}{@*)},%
%          breaklines=true,%
%          backgroundcolor=\color{thecodebackground},%
%          firstnumber=last,%
%          stepnumber=1,%
%          upquote=true,%
%          alsoletter={_,:},%
%          commentstyle=\bfseries\color{black!90},%
%          stringstyle = \color{black!90},
%          emphstyle=\verbatimfont\bfseries\color{black!80},%
%          keywordstyle= \bfseries\color{black!80},%
%          },
%      % added for better control
%      arc=0pt,
%      outer arc=0pt,
%      luaexp1/.code 2 args={\refstepcounter{texexp}\label{#2}}%Reference
%     \pgfkeysalso{luacode, enhanced, breakable, title={Example \thetexexp\ #1}},
%}
%\newenvironment{luaexp1}[1]{\tcblisting{luacode,#1}}{\endtcblisting}
%
%\newenvironment{luaexample}[3][]{\noindent\tcblisting{luaexp1={#2}{#3},#1}}%
%    {\endtcblisting}
%%
```

The following demonstrates the usage.

```
Example 9.1 atest

% \def\demomacro{Hello World!}
%
```

Example: A TesttestThis is a comment?

C.10 makeidx

We check that `\printindex` has been defined. If the test is positive then an indexing package has been loaded, otherwise we load the **makeidx**[2].

```
\ExplSyntaxOn
\cs_if_exist:cTF {printindex}
  { }
  {
    \RequirePackage{makeidx}[2000/03/29]
  }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_set:Npn \cvaref#1{\textcolor{\phdd_key_color_command_tl}{#1}}
\cs_set:Npn \colOpt#1{\textcolor{\phdkv@col@opt}{#1}}
\ExplSyntaxOff
\lstdefinestyle{tcbdocumentation}{language={[LaTeX]TeX},
    aboveskip={0\p@ \@plus 6\p@},
    belowskip={0\p@ \@plus 6\p@},
    columns=fixed,
    keepspaces=true,
    breaklines=true,
    prebreak={\Righttorque},
    postbreak={\space\Lefttorque},
    breakatwhitespace=true,
    basicstyle=\ttfamily\footnotesize,
    extendedchars=true,
    nolol,
    inputencoding = \phdkv@listingencoding}
```

C.11 Documenting Macros

The following macros are taken from **ltxdoc** and modified accordingly.

`\begin{docCommand}`[⟨*options*⟩]{⟨*name*⟩}{⟨*parameters*⟩}
⟨*command description*⟩
`\end{docCommand}`

```
\DeclareRobustCommand\phdcs[1]{{\color{thecs}{%
          \texttt{\char`\\\detokenize{#1}}}}}
```

[2] **makeidx**.

`\phddoc_orig_meta`

```
\ExplSyntaxOn
\cs_gset_eq:NN \phd@doc@orig@meta\meta%
\ExplSyntaxOff
```

(*End definition for* `\phddoc_orig_meta`.)

`\meta`{⟨*argument*⟩}

We modify the standard ⟨*arguments*⟩ to allow for colour settings. The `themeta` is defined in the **phd-colorpalette** package. The braces are not colored as they do not look very good if they are.

Use the original **meta** as defined in **doc** and only colorize and set the font parameters through the **doc** provided macro hook `\meta@font@select`.

```
\ExplSyntaxOn
\cs_set:Npn \meta #1 {
   \group_begin:
   \def\meta@font@select{\rmfamily\upshape\color{themeta}}
   \phd@doc@orig@meta{#1}
   \group_end:
}
```

`\marg`{⟨*mandatory argument*⟩}

Typesets a mandatory argument as ⟨*argument*⟩.

```
\cs_set:Npn \marg #1
  {
    \group_begin:
      \ttfamily\char`\{
      \def\meta@font@select{\rmfamily\itshape\color{theoarg}}
      \phd@doc@orig@meta{#1}
      \ttfamily\char`\}
    \group_end:
  }
```

`\oarg`{⟨*argument*⟩}

Typesets an optional argument, as found in LaTeX commands. The command:

`\test\oarg{style=two}\marg{mandatory arguments}`

will typeset:

`\test`[⟨*style=two*⟩]{⟨*mandatory arguments*⟩}

```
\cs_set:Npn \oarg #1
  {
    {\ttfamily[}\meta{#1}{\ttfamily]}
  }
\ExplSyntaxOff
```

`\docCounter`{⟨*name*⟩} New: 2016-01-08

Documents a counter with given ⟨*name*⟩. The counter is automatically indexed.

```
%The counter \docCounter{foocounter} can be used for computation.
%
```

The counter foocounter can be used for computation.

\docCounter {⟨name⟩} New: 2015-10-08

Identical to ??[→ P. ??], but without index entry.

\phdcs {⟨macro name⟩} Updated: 2017-10-11

We modify the standard \cs and save to a new name to be able to use underscores. Maybe there are better ways of doing it as well.

```
\newif\ifphd@doc@toindex
\newif\ifphd@doc@colorize
\newif\ifphd@doc@annotate
```

C.11.1 Keys for language related texts.

```
%
 \ExplSyntaxOn
\cxset{
  pageshort/.store~in           = \phdkv@text@pageshort,
  doclang/.cd,
  color/.store~in               = \phdkv@text@color,
  colors/.store~in              = \phdkv@text@colors,
  environment~content/.store~in = \phdkv@text@envcontent,
  environment/.store~in         = \phdkv@text@env,
  environments/.store~in        = \phdkv@text@envs,
  key/.store~in                 = \phdkv@text@key,
  keys/.store~in                = \phdkv@text@keys,
  index/.store~in               = \phdkv@text@index,
  pageshort/.store~in           = \phdkv@text@pageshort,
  value/.store~in               = \phdkv@text@value,
  values/.store~in              = \phdkv@text@values,
  % List Styles text in english
  lst~styles/.store~in          = \phd@text@lststyles,
}
\cxset{doclang/lst~styles       = Listings~Styles}
\ExplSyntaxOff
\cxset
  {
    documentation listing options/.store in=\phdkv@doclistingoptions,%
    documentation listing style/.style={documentation listing options={style=#1}},%
    documentation minted style/.store in=\phdkv@docmintstyle,
    documentation minted options/.store in=\phdkv@docmintoptions,
  }
```

C.12 Index settings

We consider indices to be composed of three elements, the index heading i.e., the word Index typeset in a specific language, the entries and the page numbers.

The following keys relate to settings that must have a one to one relationship with the settings of the **.ist** file. Unfortunately there is no easy way to achieve this. A better strategy is to generate the .ist file automatically by writing the parameters to a file. 16/08/2017

\tex
gobble

```
% Getting codeline and listings number at same distance
  \cxset{gobble=1} gobbles all
```

```
% Getting codeline and listings number at same distance
  \cxset{gobble=1} gobbles all
```

(*End definition for* `\tex` *and* `gobble`.)

idx@actual
\idx@quote
\idx@level
\idx@format
\idx@encap

```
\cxset{
    % indexing
    index actual/.store in   = \idx@actual,
    index quote/.store in    = \idx@quote,
    index level/.store in    = \idx@level,
    index format/.store in   = \idx@format,
    index encap/.store in    = \idx@encap,
    index colorize/.is if    = phd@doc@colorize,%
    index annotate/.is if    = phd@doc@annotate,%
  }
```

(*End definition for* `idx@actual` *and others*.)

We provide three default styles. The first style is suitable for documenting normal text without any commands or fancy characters in the index. The second is for the german language. These settings are also suitable for any language that has the `''` messed up by Babel or the @ has its catcode changed. For better mnemonics we also have one as **doc**. We default to the `//index doc settings`.

```
 \cxset{
    index default settings/.style = {index actual={@},
                                     index quote={"},
                                     index level={!}},
    index german settings/.style = {index actual={=},
                                     index quote={!},
                                     index level={>}},
    index doc settings/.style    = {index actual={=},
                                     index quote={!},
                                     index level={>}},
 }
/DOCUM
*DOCUM|DFLT
DOCUM \cxset{
   % indexing
   index actual  = {@},
   index quote   = {!},
   index level   = {>},
   index doc settings,
DOCUM }
/DOCUM|DFLT
*DOCUM
```

C.12.1 Keys for adjusting design of command displays.

The following keys relate to heading describing commands, keys environments and the like

`doc left`=⟨*length*⟩ (no default, initially 2em)

Sets the lefthand "offset" of the documentation texts for the **??**→P.**??**, **??**→P.**??**, **??**→P.**??** to ⟨*length*⟩.

`doc right`=⟨*length*⟩ (no default, initially 0em)

Sets the righthand "offset" of the documentation texts for the `\docCommand`, **??**→P.**??**, `\docKey` to ⟨*length*⟩.

```
\cxset{
    doc left/.dimstore in                   = \phdkv@doc@left,
    doc right/.dimstore in                  = \phdkv@doc@right,
    doc left indent/.dimstore in            = \phdkv@doc@indentleft,
    doc right indent/.dimstore in = \phdkv@doc@indentright,
    doc head command/.style       = {doc@head@command/.style={#1}},
    doc head environment/.style   = {doc@head@environment/.style={#1}},
    doc head key/.style           = {doc@head@key/.style={#1}},
    doc head/.style               = {doc head command= {#1},
                                     doc head environment={#1},
                                     doc head key={#1}},
    doc description/.store in     = \phdkv@doc@description,%
    doc into index/.is if         = phd@doc@toindex,%
  }
\cxset{doc description = {some test}}
```

```
% styles
\cxset{
/DOCUM
*DOCUM|DFLT
  docexample/.style={colframe=ExampleFrame,colback=ExampleBack,fontlower=\footnotesize},
  documentation minted style=,
  documentation minted options={tabsize=2,fontsize=\small},
  english language/.code={\cxset{doclang/.cd,
    color=color,colors=Colors,
    environment content=environment content,
    environment=environment,environments=Environments,
    key=key,keys=Keys,
    index=Index,
    pageshort={P.},
    value=value,values=Values}},
 }
/DOCUM|DFLT
*DOCUM
\AtBeginDocument{%
  \csname phd@doc@index@\idx@format\endcsname%
  \hypersetup{
    citecolor  = \phdkv@colhyper,
    linkcolor  = \phdkv@colhyper,
    urlcolor   = \phdkv@colhyper,
```

```
    filecolor  = \phdkv@colhyper,
    menucolor  = \phdkv@colhyper
}}

\lstdefinestyle{smalldisplay}{%
  numbers=none,
  backgroundcolor=\color{thecodebackground},
  xleftmargin=0pt}

\tcbset{documentation listing style=smalldisplay}
\tcbset{
   docexample/.style={%
   colframe=thecodeframe,
   colback=thecodebackground,
   before skip=\medskipamount,
   after skip=\medskipamount,
   fontlower=\footnotesize,
   }
}
%
%\newenvironment{dispListing*}[1]{%
%  \phd@layer@pushup\cxset{docexample,#1}\phdwritetemp%
%  }{\enddispListing}

% index auxiliary macros

\ExplSyntaxOn
\cs_set:Npn \phdindexprintca #1#2#3 {
  \ifphd@doc@colorize
    \textcolor{#2}
    { \texttt{#1} }
  \else
    \texttt{#1}
  \fi
  \ifphd@doc@annotate\
   #3
  \fi
}


\cs_set:Npn \phd_index_print_c#1#2{%
  \ifphd@doc@colorize
    \textcolor{#2}{\texttt{#1}}
  \else\texttt{#1}
  \fi%

}


\NewDocumentCommand{\phdindexprintcomc}{ m }
  {
    \phd_index_print_c {\phdcs{#1}}{\phdd_key_color_command_tl}
  }
%%

```

\phd_print_com {⟨*cs name*⟩}

Prints a command.

```
\cs_new:Npn \phd_print_com #1
  {
    \textcolor{black}{\ttfamily\bfseries\phdcs{#1}}%\phdd_key_color_command_tl
  }
\ExplSyntaxOff
\ExplSyntaxOn
\newrobustcmd{\phdindexprintenvca}[1]
  {
    \phdindexprintca{#1}{\phdd_key_color_environment_tl}{\phdkv@text@env}
  }

\newrobustcmd{\phdindexprintenvc}[1]
  {
    \phd_index_print_c{#1}{\phdd_key_color_environment_tl}
  }
\ExplSyntaxOff
\ExplSyntaxOn
\cs_set:Npn \phd_print_env#1
  {
    \textcolor{\phdd_key_color_environment_tl}{\ttfamily\bfseries#1}
  }

\newrobustcmd{\phdindexprintkeyca}[1]
  {
    \phdindexprintca{#1}{\phdd_key_color_tl}{\phdkv@text@key}
  }

\newrobustcmd{\phdindexprintkeyc}[1]{\phd_index_print_c{#1}{\phdd_key_color_tl}}



\newrobustcmd {\phdindexprintvalca}[1]
  {
    \phdindexprintca{#1}{\phdd_key_color_value_tl}{\phdkv@text@value}
  }

\newrobustcmd {\phdIndexPrintValC}[1]
  {
    \phd_index_print_c{#1}{\phdd_key_color_value_tl}
  }

\cs_set:Npn \phd@Print@Val #1
  {
    \textcolor {\phdd_key_color_value_tl} {\ttfamily\bfseries#1}
  }

\newrobustcmd{\phdindexprintcolca}[1]
  {
    \phdindexprintca{#1}{\phdd_key_color_color_tl}{\phdkv@text@color}
```

```
  }



\cs_set:Npn \phdindexcom #1
  {
    \ifphd@doc@toindex
      \index
       {
         #1
         \idx@actual
         \phdindexprintcomc{#1}
       }
    \fi
  }
```

\phd_index_env{⟨*environment name*⟩}

Auxiliary command to add an environment to the index. The environment name need to come from the key settings. This need to be integrated with the i18n package later on.

```
\cs_set:Npn \phd_index_env #1
  {
    \ifphd@doc@toindex
      \index
        {#1
          \idx@actual
          \phdindexprintenvca{#1}
        }
      \index
        {
          \phdkv@text@envs
          \idx@level#1
          \idx@actual
          \phdindexprintenvc{#1}
        }
    \fi
  }
```

\phd_index_key{⟨*key name*⟩}

Indexes a key.

```
\cs_set:Npn \phd_index_key_path #1#2
  {
    \ifphd@doc@toindex\index{#2\idx@actual
      \phdindexprintkeyca{#2}}
      \index{\phdkv@text@keys
         \idx@level#1
        \idx@actual
        \phdindexprintkeyc{/#1/}
        \idx@level#2
```

```
        \idx@actual
        \phdindexprintkeyc{#2}
      }
    \fi
  }
```

\phd_index_col {⟨*color name*⟩}

```
\cs_set:Npn \phd_index_col #1
  {
    \ifphd@doc@toindex
    \index
      {
        #1
        \idx@actual
        {\phdindexprintcolca{#1}}
        \encapchar hyperpage
      }
    \index
      {
        \phdkv@text@colors \idx@level #1
        \idx@actual{\phdindexprintcolc {#1}\protect\thinspace\protect\docColorBox{#1}}
        \encapchar hyperpage
      }
    \fi
  }

\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_set:Npn \phd_brackets #1
  {
    {\ttfamily\char`\{}#1{\ttfamily\char`\}}
  }
\ExplSyntaxOff
```

\begin{phdd_manual_entry}
⟨*environment content*⟩
\end{phdd_manual_entry}

A list enclosing manual entries.

```
\ExplSyntaxOn
\newenvironment{phdd_manual_entry}%
  {%
   \begin{list}{}
    {%
     \setlength{\leftmargin}{\phdkv@doc@left}%
     \setlength{\itemindent}{0pt}%
     \setlength{\itemsep}{0pt}%
     \setlength{\parsep}{0pt}%
     \setlength{\rightmargin}{\phdkv@doc@right}%
    }\item
  }
  {\end{list}}
```

```
\cs_set:Npn \phd_manual_top #1
  {
    \itemsep=0pt
    \parskip=0pt
    \item\strut{#1}\par
    \topsep=0pt
  }

\cs_set:Npn \phd_doc_do_description:
  {
    \ifx\phdkv@doc@description\@empty
    \else
      \tcblower
      \raggedleft(\phdkv@doc@description)
    \fi
  }
\ExplSyntaxOff
```

\begin{phd@doc@head} {⟨*additional options*⟩}
⟨*environment content*⟩
\end{phd@doc@head}

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

```
\ExplSyntaxOn
\newtcolorbox{phd@doc@head}[1]
 {
  blank,
  colback=red!10,
  colframe=white,
  code={\tcbdimto\tcb@temp@grow@left{-\phdkv@doc@indentleft}%
        \tcbdimto\tcb@temp@grow@right{-\phdkv@doc@indentright}},
  grow~to~left~by=\tcb@temp@grow@left,%
  grow~to~right~by=\tcb@temp@grow@right,
  sidebyside,
  sidebyside~align=top,
  sidebyside~gap=-\tcb@w@upper@real,
  phantom=\phantomsection,%
  enlarge~bottom~by=-0.2\baselineskip,
  #1
 }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\newenvironment{docCmd}[3][]{
  \cxset{#1}%
  \begin{phdd_manual_entry}%
  \begin{phd@doc@head}{doc@head@command}%
  \phd_print_com{#2}
  \phdindexcom{#2}
  \protected@edef\@currentlabel{\noexpand\phdcs{#2}}
```

```
  \label{com:#2}{\ttfamily #3}%
  \phd_doc_do_description:%
  \end{phd@doc@head}}%
  {\end{phdd_manual_entry}}
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\newenvironment{docCmd*}
  {
    \begingroup
    \phd@doc@toindexfalse
    \begin{docCommand}
  }
  {
    \end{docCommand}
    \endgroup
  }

\newenvironment{docEnv}[3][]{\cxset{#1}%
  \begin{phdd_manual_entry}%
  \begin{phd@doc@head}{doc@head@environment}%
  \strut
  \phdcs{begin}
  \phd_brackets{\phd_print_env{#2}}
  \phd_index_env{#2}
  \protected@edef\@currentlabel{#2}\label{env:#2}{\ttfamily #3}%\par%
  \strut~~\meta{\phdkv@text@envcontent}%\par%
  \strut\phdcs{end}
  \phd_brackets{\phd_print_env{#2}}%
  \phd_doc_do_description:%
  \end{phd@doc@head}}%
  {\end{phdd_manual_entry}}
\ExplSyntaxOff
```

`\begin{docEnv*}`⟨*environment content*⟩`\end{docEnv*}`

```
\newenvironment{docEnv*}
  {
    \bgroup
    \phd@doc@toindexfalse
    \begin{docEnv}
  }
  { \end{docEnv}\egroup }
```

`\begin{docKey}`[⟨*key path*⟩][⟨*options*⟩]{⟨*name*⟩}{⟨*parameters*⟩}{⟨*description*⟩}
⟨*key description*⟩
`\end{docKey}`

`/foo/footitle=`⟨*text*⟩ (no default, initially empty)

Creates a heading line with ⟨*text*⟩ as content.

Example 12.2 Using dockey

```
%\begin{docKey}[foo]{footitle}{ = \meta{text}}{no default, initially empty}
%  Creates a heading line with \meta{text} as content.
%\end{docKey}
%
```

/foo/footitle = ⟨*text*⟩ (no default, initially empty)

Creates a heading line with ⟨*text*⟩ as content.

```
\ExplSyntaxOn
\renewenvironment{docKey}[4][\@empty]{
  \begin{phdd_manual_entry}%
    \cxset{doc~description={#4}}%
```

The contents of the key are placed using **\phd@doc@head**, which is a tcolorbox. **doc@head@key** is a style for the tcolorbox.

```
      \begin{phd@doc@head}{doc@head@key}
        \ifx#1\@empty%
          \phd_print_key:n {#2}
          \phd_index_key{#2}
          \protected@edef\@currentlabel{#2}
          \label{key:#2}{\ttfamily #3}%
        \else
          \phd_print_key:n {/#1/#2}
          \phd_index_key_path{#1}{#2}
          % redefine current label
          \protected@edef\@currentlabel{/#1/#2}
          % label
          \label{key:/#1/#2}{\ttfamily#3}%
        \fi%
     \phd_doc_do_description:
     \end{phd@doc@head}}
  {\end{phdd_manual_entry}}

\renewenvironment{docKey*}
  {\bgroup\phd@doc@toindexfalse\begin{docKey}}
  {\end{docKey}\egroup}

\cs_set:Npn \phdmakedocSubKey#1#2{%
  \newenvironment{#1}[4][\@empty]{%
    \ifx##1\@empty
      \cs_set:Npn \phd@key@path {#2}
    \else
     \cs_set:Npn \phd@key@path{#2/##1}
    \fi%
    \begin{docKey}[\phd@key@path]{##2}{##3}{##4}}%
    {\end{docKey}}%
  \newenvironment{#1*}{\bgroup\phd@doc@toindexfalse\begin{#1}}{\end{#1}\egroup}%
}
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_set:Npn \@@_doc_aux_env: #1
  {
    \phd_print_env{#1}
    \phd_index_env{#1}
  }

\cs_set:Npn \@@_doc_aux_env_star #1
  {
    \phd_print_env{#1}
  }
```

\docAuxEnv ⟨*star*⟩ {⟨*arg1*⟩}

```
\NewDocumentCommand\docAuxEnv { s m }
  {
    \IfBooleanTF {#1}
    {\@@_doc_aux_env_star{#2} }
    {\@@_doc_aux_env: {#2} }
  }
\ExplSyntaxOff
```

\docAuxKey[⟨*path*⟩]{⟨*value*⟩}

Prints in-line the value of a key. For pgf keys the optional argument typesets the family:

Example: `\docAuxKey[phd]{index doc settings}` typesets: `/phdd/index doc settings`

\phd_print_key:n
\phd_index_key:n
\phdd_index_key:n

Besides the main function we require a function to typeset the key another on to typeset the index and auxiliaries to handle the star and optional arument.

```
\ExplSyntaxOn
\cs_set:Npn \phd_print_key:n #1
  {
    \textcolor{\phdd_key_color_tl}{\ttfamily\bfseries#1}
  }

\cs_set:Npn \phd_index_key #1
  {
    \ifphd@doc@toindex
      \index{#1\idx@actual \phdindexprintkeyca{#1}}
      \index
        {
          \phdkv@text@keys
          \idx@level
          #1
          \idx@actual
          \phdindexprintkeyc{#1}
        }
    \fi
  }

\NewDocumentCommand {\@@_doc_aux_key:} {O{\@empty} m}
```

```
      {%
         \ifx#1\@empty%
           \phd_print_key:n {#2}
           \phd_index_key {#2}%
         \else%
           \phd_print_key:n {/#1/#2}
           \phd_index_key_path {#1}{#2}%
      \fi
      }%

    \newcommand{\@@_doc_aux_key_star}[2][\@empty]{%
      \ifx#1\@empty%
       \phd_print_key {#2}%
      \else%
       \phd_print_key {/#1/#2}%
      \fi}%

    \DeclareDocumentCommand {\docAuxKey} { s O{} m }
      {
        \IfBooleanTF {#1}
          { \@@_doc_aux_key_star {#3} }
          { \@@_doc_aux_key: [#2]{#3} }
      }
    \ExplSyntaxOff
```

(*End definition for* `\phd_print_key:n`, `\phd_index_key:n`, *and* `\phdd_index_key:n`.)

`\phd_ref_doc` {⟨*refname*⟩}

We use **hyperref**[3] to add links. The `\hyperref`[⟨*label*⟩]{⟨*text*⟩} is used to create the link. We use `\ding{213}` → for the page see `\docColor`. This is a great technique pioneered in the PGF and PGFPlots manuals for cross referencing and the code below is an adaptation.

There are two colors involved here and for a good reason. The first color needs to emphasize the command and to be consistent with the description macros. The second is the hyperlink color. For very long manuals, this can get a bit ugly, so one may want to de-emphasize it, like I have done here.

`\setrefcountdefault`
`\getpagerefnumber`

References are not numbers, however they often store numerical data such as section or page numbers. `\ref` or `\pageref` cannot be used for counter assignments and calculations because they are not expandable. The package **refcount**[4] can be used to extract the data from references. The package supports `hyperref`, `nameref`, `titleref`, and `babel` so it is safe to use. We set the default value here. `\docColors`

```
\ExplSyntaxOn
\setrefcountdefault{-1}
```

(*End definition for* `\setrefcountdefault` *and* `\getpagerefnumber`.)

```
\cs_set:Npn \phd_ref_doc #1 #2
{
```

[3] **hyperref**.
[4] **refcount**.

```
  \IfRefUndefinedExpandable {#2}
    {{\bfseries\phdd_typeset_command {#1} }}
    {
      \hyperref[#2]
      {{\color{themacrocolor}\texttt{\ref*{#2}}}%
      \ifnum\getpagerefnumber{#2}=\thepage
      \else
        \textsuperscript
      {
        \ding{213}\,
        \phdkv@text@pageshort\,
        \pageref*{#2}
      }
     \fi
   }
  }
 }

\cs_set:Npn \phd_ref_doc_star#1
  {
    \hyperref[#1]{\texttt{\ref*{#1}}}
  }
\ExplSyntaxOff
```

\refCmd⟨*⟩ {⟨*name*⟩}

The \refCmd references a documented macro, where name is written without a backslash. the page reference is supressed if it links to the same page. The star version does not print a page reference.

\ref_com:

```
\ExplSyntaxOn
\cs_set:Npn \ref_com: #1
  {
    \phd_ref_doc {#1} {com:#1}
  }

\cs_set:Npn \ref_com_star #1
  {
    \phd_ref_doc_star{com:#1}
  }
```

(*End definition for* \ref_com:.)

```
\DeclareDocumentCommand {\refCmd} { s m }
  {
    \IfBooleanTF {#1}
    { \ref_com_star {#2} } { \ref_com: {#2} }
  }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_set:Npn \refEnv: #1 {\phd_ref_doc{env:#1}}
\cs_set:Npn \refEnv_star#1{\phd_ref_doc_star{env:#1}}

```

```
\DeclareDocumentCommand {\refEnv} { s m }
  {
    \IfBooleanTF {#1}
      { \refEnv_star {#2} }
      { \refEnv: {#2}     }
  }
\ExplSyntaxOff
```

\refKey*{⟨*ref text*⟩}

```
\ExplSyntaxOn
\cs_set:Npn \refKey@#1{\phd_ref_doc{key:#1}}
\cs_set:Npn \refKey@star#1{\phd_ref_doc_star{key:#1}}
\DeclareDocumentCommand {\refKey} { s m }
  {
    \IfBooleanTF { #1 }
      { \refKey@star {#2} }
      { \refKey@ {#2}       }
  }
\ExplSyntaxOff
```

\refAux{⟨*command name*⟩}

```
\ExplSyntaxOn
\cs_set:Npn \refAux#1{\textcolor{\phdkv@colhyper}{\ttfamily #1}}
\cs_set:Npn \refAuxcs#1{\textcolor{\phdkv@colhyper}{\phdcs{#1}}}
\ExplSyntaxOff
```

C.13 Indexing

Most of the indexing macros that follow have been adapted from the pgfmanual-en-macros or the tcolorbox documentation code and transliterated to expl3 language.

```
\ExplSyntaxOn
\cs_set:Npn \phd@doc@index@pgf@
  {
    \c@IndexColumns=2%
    \cs_set:Npn \theindex
      {
        \@restonecoltrue
        \columnseprule 0pt
        \columnsep 28\p@
        \twocolumn[\index@prologue]%
        \parindent -30pt%
        \columnsep 15pt%
        \parskip 0pt plus 1pt%
        \leftskip 30pt%
        \rightskip 0pt plus 2cm%
        \small%
        \cs_set:Npn \@idxitem{\par}%
        \let\item\@idxitem\ignorespaces
      }
```

```
    \cs_set:Npn \endtheindex{\onecolumn}%
    \cs_set:Npn \noindexing{\let\index=\@gobble}%
  }
```

C.13.1 Index heading and prologue

The index prologue is text that is entered just before the indexing entries start. Most indices do not have any text. We also need to distinguish between using a chapter type heading or a section type heading.

```
\cs_set:Npn \phd@doc@index@pgfsection{%
  \cs_set:Npn \index@prologue
    {
      \section*{\phdkv@text@index}
      \addcontentsline{toc}{section}{\phdkv@text@index}
      \par\noindent%
    }
  \phd@doc@index@pgf@
}
\cs_set:Npn \phd@doc@index@pgfchapter{%
  \cs_set:Npn \index@prologue{\ifdefined\phantomsection\phantomsection\fi
  \@makeschapterhead{\phdkv@text@index}
  \addcontentsline{toc}{chapter}{\phdkv@text@index}}%
  \phd@doc@index@pgf@%
}
%
\let\phd@doc@index@pgf=\phd@doc@index@pgfsection%
\cs_set:Npn \phd@doc@index@doc
  {
    \let\phdindexcom      = \SpecialMainIndex
    \let\phd_index_env    = \SpecialMainEnvIndex
    \cxset{index german settings}
    \EnableCrossrefs
    \PageIndex
}

\cs_set:Npn \phd@doc@index@off{}%
\ExplSyntaxOff
\tcbset{%
  reset@documentation/.style={%
    },
  documentation listing style=tcbdocumentation,
  doc left=2em,
  doc right=0pt,
  doc left indent=-2em,
  doc right indent=0pt,
  doc head=,
  before doc body=,
  after doc body=,
  doc description=,
  doc into index=true,
  doc marginnote=,
```

```
  initialize@reset=reset@documentation,
  color option=Option,
  color definition=Definition,
  color hyperlink=Hyperlink,
  index command=\index,%
  index format=pgf,
  index default settings,
  index colorize=false,
  index annotate=true,
  english language,
  keywords bold,
}
\cxset{%
    reset@documentation/.style={%
    index format=pgf,
    english language,
    documentation listing style = tcbdocumentation,
    index default settings,
    color option=Option,
    color definition=Definition,
    color hyperlink=Hyperlink,
    before example=\par\smallskip,
    after example=,
    doc left=0em,
    doc right=0pt,
    doc left indent=-2em,
    doc right indent=0pt,
    doc head=,
    doc description=,
    doc into index=true,
    index colorize = true,
    index annotate= false,
    },
%  initialize@reset=reset@documentation,
}
\cxset{reset@documentation}
```

We set the `//index format=pgf` and the rest of the keys to the **`german`** settings that are suitable for **`doc`**.

```
\cxset{index format  =  pgfchapter,
       index actual={=},
       index level = {>},
       index quote = {!},
       index german settings,
       color hyperlink = thelinkcolor,  % links with color palette
       color definition =thelinkcolor,  % links with color palette
       pageshort       = {$\sigma{}$},
   }
```

`\main` This is defined in doc and is the user command for styling the main page in an index.

```
\def\main#1{{\color{red}\uline{#1}}}
```

(*End definition for* `\main`.)

C.14 Miscellaneous doc commands

For consistency all commands that typeset their content, as well as index it and perhaps, also reference it have the prefix **doc**.

\docFile {⟨*file name*⟩}

Typesets and index a file.

Example: **template_html.tpl**

```
\ExplSyntaxOn
\cs_set:Npn \@@_print_docfile:n #1 {%
    \begingroup
    \color{thered}
    \ttfamily {\tl_to_str:n{#1}}
    \endgroup
}

\cs_set:Npn \docFile #1
  {
    \begingroup
      \@@_print_docfile:n {#1}
    \endgroup
    \index{files:\idx@level\protect\detokenize{#1}\encapchar hyperpage}
  }
```

\docExtension Extension macro

```
\cs_set:Npn \docExtension #1
  {
    \texttt {#1}
    \index{file~extensions\idx@level#1\encapchar hyperpage}
  }
\let\docextension\docExtension
\ExplSyntaxOff
```

(*End definition for* `\docExtension`.)

C.15 Other Indexing functions

\indexmany [⟨*category*⟩] {⟨*clist*⟩}

This function indexes a comma delimited list of items. It is convenient when you have paragraphs with a lot of terms.

```
 \ExplSyntaxOn
 \DeclareDocumentCommand\indexmany {o m }
 {
   \clist_gset:Nn \indexmany: {#2}
   \IfValueTF {#1}
    {
      \clist_map_inline:Nn\indexmany:
        {
          \index{#1\idx@level##1}\index{##1}
        }
```

```
    }
    {
     \clist_map_inline:Nn\indexmany:
      {
        \index{##1}
      }
    }
 }
 \ExplSyntaxOff
```

\idxlanguage{⟨*language name*⟩}

Typesets and index a lanaguage name. Used in the i18n package.

Example: : `idxlanguage{cypriote}` `cypriote`

```
\DeclareRobustCommand{\idxlanguage}[1]{%
  \index{\string #1 (script)}
  \index{scripts:\idx@level#1\encapchar hyperpage}
  \texttt{#1}\xspace}%
```

C.16 Indexing Symbols

The commands that are defined in this section can be used to produce almost any symbol that is described in the Comprehensive Symbols[5]. They are mostly variants as described by this publication. Some modernized a bit, others using LaTeX 3 type definitions.

Some of them are just used for typesetting symbols in tables.

\indexboth {⟨*arg1*⟩} {⟨*arg2*⟩}

Indexes both arguments.

Table C.1: **textcomp**
Currency Symbols

	\textbaht	$	\textdollar*	₲	\textguarani		\textwon
¢	\textcent	$	\textdollar	£	\textlira	¥	\textyen
¢	\textcent		\textdong		\textnaira		
	\textcolonmonetary	€	\texteuro		\textpeso		
¤	\textcurrency	ƒ	\textflorin	£	\textsterling*		

```
\newcommand{\idxboth}[2]{\mbox{}\index{#1 #2}\index{#2\idx@level#1}}
\newcommand{\idxbothbegin}[2]{\mbox{}\index{#1 #2|(}\index{#2\idx@level #1|(}}
\newcommand{\idxbothend}[2]{\mbox{}\index{#1 #2|)}\index{#2\idx@level #1|)}}
```

[5] **comprehensive**.

```
\ExplSyntaxOn
\cs_gset_eq:NN \indexboth\idxboth
\cs_gset_eq:NN \indexbothbegin \idxbothbegin
\cs_gset_eq:NN \indexbothend\idxbothend
\ExplSyntaxOff
```

We define a related macro for indexing accents. In a previous version of this file, **\indexaccent** additionally included "see also accents" in the index. This became distracting so I made **\indexaccent** a synonym for **\indexcommand** for the time being. Because punctuation marks can be problematic for makeindex, we define an acro that sorts its argument under the comparatively innocuous "**_**".

```
\def\cmd#1{\texttt{\string#1}\indexcommand{#1}}
\newcommand{\cmdI}[2][]{%
  \def\first@arg{#1}%
  \ifx\first@arg\@empty
    \texttt{\string#2}\indexcommand[#2]{#2}%
  \else
    \texttt{\string#2}\indexcommand[#1]{#2}%
  \fi
}
\newcommand{\cmdX}[1]{\cmdI[$\string#1$]{#1}}
\newcommand{\cmdW}[1]{\cmdI[$\string\blackacc{\string#1}$]{#1}}
\newcommand{\cmdIp}[2][]{%
  \def\first@arg{#1}%
  \ifx\first@arg\@empty
    \texttt{\string#2}\indexpunct[#2]{#2}%
  \else
    \texttt{\string#2}\indexpunct[#1]{#2}%
  \fi
}
```

\sanitize{⟨*command*⟩}

Delimited macro (!!!) that sanitizes macros. Classic TeXBook style. It removes the backslash. Not too sure if the word sanitize is appropriate. Another approach is to gobble the backslash. See **??** → P. **??**.

```
\begingroup
 \catcode`\|=0
 \catcode`\\=12
 |gdef|sanitize#1#2!!!{%
   |ifx#1\%
     #2%
   |else
     #1#2%
   |fi
}
|endgroup
```

\indexcommand[⟨⟩]{⟨*command*⟩}

Index a *symbol*, which may or may not begin with a *backslash*. (Is there a better way to do this?) Also, if symbol is given as an optional argument is given, typeset that symbol in the index, as well

```
 \ExplSyntaxOn
\NewDocumentCommand \indexcommand { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    %\def\first@arg{#1}%
    \IfNoValueTF{#1}
    {
       \exp_after:wN
          \index {
             \sanitized
             \idx@actual
             {\expandafter
             \phdIndexPrintCs{\sanitized}}
             \encapchar~hyperpage
             }%
    }
    {
       \expandafter
         \index\expandafter{\sanitized\idx@actual{\string\verb+\string#2+ (#1)}\encapcha
    }
  }
 \ExplSyntaxOff

```

C.16.1 Indexing archaic scripts.

\indexcypriot [⟨*arg1*⟩] {⟨*arg2*⟩}

Index helper function for indexing Cypriot script. Only used in the phd documentation.

```
\NewDocumentCommand \indexcypriot { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \expandafter\index\expandafter{Cypriot\idx@level\sanitized=\string\verb+\string#2
    }
    {
       \expandafter\index\expandafter{Cypriot>\sanitized=\string\verb+\string#2+ (#1)}%
    }
  }
```

\indexstaves [⟨*arg1*⟩] {⟨*arg2*⟩}

Index helper function for indexing Icelandic staves. Only used in the phd documentation.

```
\NewDocumentCommand \indexstaves { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \expandafter\index\expandafter{Staves>\sanitized=\string\verb+\string#2+}%
    }
```

```
    {
       \expandafter\index\expandafter{Staves>\sanitized=\string\verb+\string#2+ (#1)}%
    }
  }
```

\indexlinearb [⟨*typesetting command(s)*⟩] {⟨*command*⟩}

Index helper function for indexing the linearb script. Only used in the phd documentation for scripts. In .idx it writes

```
%    \indexentry{Linear B >BPamphora=\phdIndexPrint {BPamphora} (\textlinb
%
```

\𐀀

```
\newrobustcmd\phdIndexPrintCs[1]{{%
  \catcode`\_=12\relax\catcode`\@11\relax\ttfamily\char`\\\scantokens{#1}\unskip}}
%
%
\NewDocumentCommand \indexlinearb { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \index{Linear B\idx@level\string#1\idx@actual
        {\string\verb+\string#2+}\encapchar hyperpage}
    }
    { % index
         \index{Linear B
         \idx@level
         \string #1
         \idx@actual
         {\phdIndexPrintCs{\sanitized} (#1)}\encapchar hyperpage}%
       % typeset
       \string#2 & (#1)
    }
  }
```

\BPamphora	(𐀀)	\BPchassis	(𐀁)	\BPman	(𐀂)	\BPwheat	(𐀃)
\BParrow	(𐀄)	\BPcloth	(𐀅)	\BPnanny	(𐀆)	\BPwheel	(𐀇)
\BPbarley	(𐀈)	\BPcow	(𐀉)	\BPolive	(𐀊)	\BPwine	(𐀋)
\BPbilly	(𐀌)	\BPcup	(𐀍)	\BPox	(𐀎)	\BPwineiih	(𐀏)
\BPboar	(𐀐)	\BPewe	(𐀑)	\BPpig	(𐀒)	\BPwineiiih	(𐀓)
\BPbronze	(𐀔)	\BPfoal	(𐀕)	\BPram	(𐀖)	\BPwineivh	(𐀗)
\BPbull	(𐀘)	\BPgoat	(𐀙)	\BPsheep	(𐀚)	\BPwoman	(𐀛)
\BPcauldroni	(𐀜)	\BPgoblet	(𐀝)	\BPsow	(𐀞)	\BPwool	(𐀟)
\BPcauldronii	(𐀠)	\BPgold	(𐀡)	\BPspear	(𐀢)		
\BPchariot	(𐀣)	\BPhorse	(𐀤)	\BPsword	(𐀥)		

\indexugar [⟨*arg1*⟩] {⟨*arg2*⟩}

Index helper function for indexing Ugaritic scripts. Only used in the phd

documentation.

```
\NewDocumentCommand \indexugar { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \expandafter\index\expandafter{Ugarite>\sanitized=\string\verb+\string#2+}%
    }
    {
       \expandafter\index\expandafter{Ugarite>\sanitized=\string\verb+\string#2+ (#1)}%
    }
  }
```

\indexvarpersian [⟨*arg1*⟩] {⟨*arg2*⟩}

Indexing and doc command for var Persian tables.

```
\NewDocumentCommand \indexvarpersian { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \expandafter\index\expandafter{var Persian>\sanitized=\string\verb+\string#2+}%
    }
    {
       \expandafter\index\expandafter{var Persian>\sanitized=\string\verb+\string#2+ (#1
    }
  }
```

\indexsoutharabian[⟨⟩]{⟨⟩}

Indexing and doc command for symbols tables.

```
\NewDocumentCommand \indexsoutharabian { o m }
  {
    \edef\sanitized{\expandafter\sanitize\string#2!!!}%
    \IfNoValueTF{#1}
    {
       \expandafter\index\expandafter{South Arabian>\sanitized=\string\verb+\string#2+}%
    }
    {
       \expandafter\index\expandafter{South Arabian>\sanitized=\string\verb+\string#2+ (
    }
  }
```

C.16.2 Indexing mathematical symbols

The currently available fonts The following indexing commands are auxiliary commands to index unicode symbols for maths. 26-06-2015

```
\NewDocumentCommand \indexmathcmd { o m }
  { \def\cmd@to@cs#1#2{\char\number`#2\relax}
    \IfNoValueTF{#1}
    {
```

```
        \index{\string #2\idx@actual{\string\verb+\string#2+
        ($#2$)}\encapchar hyperpage}
        % put command also
      \index{\string#1\idx@actual{\string\verb+\string#2+ ($\string#2$)*}\encapchar hype
    }
    {
      % with index level
      \index{#1\idx@level\protect\detokenize{#2}
            \idx@actual{\protect\detokenize{#2}($#2$)}\encapchar hyperpage}%
      % command alone an symbol in parentheses
      \index{\cmd@to@cs#2\idx@actual{\string\verb+\string#2+ ($\string#2$)}\encapchar hy
    }
  }
```

\indexaccent

Syntactic sugar identical to \indexcommand

```
\ExplSyntaxOn
\cs_gset_eq:NN \indexaccent\indexcommand
\cs_new:Npn \CLSLpipe {|}
\ExplSyntaxOff
```

\indexpunct [⟨*arg1*⟩] {⟨*arg2*⟩}

Indexing punctuation marks for latin scripts.

```
  \newcommand{\indexpunct}[2][]{%
    \def\first@arg{#1}%
    \def\second@arg{#2}%
    \ifx\first@arg\@empty
      \ifx\second@arg\CLSLpipe
        \index{_=\magicvertname}%
      \else
        \index{_=\string\verb+\string#2+}%
      \fi
    \else
      \ifx\second@arg\CLSLpipe
        \index{_=\magicvertname{} (#1)}%
      \else
        \index{_=\string\verb+\string#2+ (#1)}%
      \fi
    \fi
  }
```

```
%\usepackage{longdiv}
\newcommand\FC{\pkgname{fc}}
\newcommand\VIET{\pkgname{vietnam}}
%\newcommand\ABX{\pkgname{mathabx}}
```

\incsyms ⟨*void*⟩

We define an integer counter **totalsymbols** to keep track of all the symbols we load and list.[6] These are symbols which can be produced using command sequences.

[6] Unicode characters are counted separately and are dealt with under the phd-scriptsmanager package.

Define a counter to keep track of how many symbols are listed. Output this counter to the log file at the end of each run. Define **\prevtotalsymbols** to be the total number of symbols from the previous run.

```
\ExplSyntaxOn
  \int_new:c {totalsymbols}
  \cs_new:Npn \incsyms { \int_gincr:c {totalsymbols} }
  \cs_new:Npn \thetotalsymbols {\int_use:c {totalsymbols} }
\ExplSyntaxOff
```

\graybox ⟨*void*⟩

```
\newcommand*{\graybox}{\textcolor{thecodebackground}{\rule[-
\adp]{\awd}{\aht}}}

% Define |\blackacc| to display an accented box, given an accent command.
% Define |\blackacchack| to display an accented "a" and then black out
% the "a".
\newlength\awd
\newlength\aht
\newlength\adp
\settowidth{\awd}{\normalfont a}
\settoheight{\aht}{\normalfont a}
\settodepth{\adp}{\normalfont a}
\advance\adp by 0.06pt    % In Computer Modern, "a" extends slightly below its bounding
\advance\aht by \adp


\gdef\blackacchack#1{#1a\llap{\graybox}}
\gdef\blackacc#1{#1{\graybox}}
\gdef\blackacctwo#1{#1{\graybox}{\graybox}}
```

Symbol+verbatim for various types of symbols

```
\def\E#1{%
  \begingroup
    \lccode`|=`\\
    \def\EStruename{ES#1T}
    \lowercase{\incsyms\index{#1=\string\verb+\string|#1+ (\string|\EStruename)}}
  \endgroup
  \csname ES#1T\endcsname
  & \csname ES#1D\endcsname
  &
  \ttfamily\expandafter\string\csname#1\endcsname
}
```

These commands are here to be able to index these symbols for the index and to typeset them in the symbols appendix.

\Kcyp[⟨*text cmd*⟩] {⟨*symbol command*⟩}

Indexes and prints the Cypriot archaic font symbols.

Example: **\Kcyp[\textcypr{\Ca}]\Ca**

```
\def\Kcyp@opt@arg[#1]#2{\incsyms\indexcypriot[\textcypr{#1}]{#2}#1 &\ttfamily\string#2}
\def\Kcyp@no@opt@arg#1{\incsyms\indexcypriot[\textcypr{#1}]{#1}#1 &\ttfamily\string#1}
\def\Kcyp{\@ifnextchar[{\Kcyp@opt@arg}{\Kcyp@no@opt@arg}}
```

\Kstav [⟨*cmd*⟩] {⟨*stave cmd*⟩}

Indexes and prints an Icelandic stave.

```
\ExplSyntaxOn

\cs_set:Npn \Kstav_opt_arg [#1]#2
  {
    \incsyms\indexstaves[#1]{#2}# 1 &\ttfamily\string#2
  }

\cs_set:Npn \Kstav_no_opt_arg #1
  {
    \incsyms\indexstaves[#1]{#1}#1 &\ttfamily\string#1
  }

\NewDocumentCommand\Kstav {o m} {
  \IfNoValueTF {#1}
    {
      \Kstav_no_opt_arg {#2}
    }
    {
      \Kstav_opt_arg [#1] {#2}
    }
}
\ExplSyntaxOff
```

\K [⟨⟩] {⟨*cmd*⟩}

Adds a symbol cmd to a table and the index.

```
\ExplSyntaxOn
\cs_set:Npn \K@opt@arg#1#2
   {
      \incsyms
      \indexcommand[#1]{#2}#1 &\footnotesize\ttfamily\string#2
   }

\cs_set:Npn \K@no@opt@arg#1
  {
    \incsyms
    \indexcommand[#1]{#1}#1 &\footnotesize\ttfamily\string#1
  }
```

\K[⟨*symbol command*⟩] {⟨*symbol command*⟩}

Command used to typeset a table of symbols such as textsymbols. The mandatory argument takes a command such as **texteuro**, typesets the symbol first and then the command. It also adds it to the index. It is the most common command and hence we keep it short. It was originally defined in Comprehensive[7]. It also increases the counter **incsyms**. The optional argument has an alternative symbol.

Example: **\K\Square** `\Square`

[7] **comprehensive**.

```
\NewDocumentCommand {\K} { o m }
{
  \IfNoValueTF {#1} { \K@no@opt@arg {#2} } {\K@opt@arg {#1}{#2}}
}
\ExplSyntaxOff
```

Table C.3: LaTeX Escapable "Special" Characters

$	\$	%	\%	_	_*	}	\}	&	\&	#	\#	{	\{

```
\ExplSyntaxOn
\cs_set:Npn \Kp #1
   {
       \incsyms
       \indexpunct[$#1$]{#1}#1 &\footnotesize\ttfamily\string#1
    }
\ExplSyntaxOff
%\begin{symtable}[EDICE]{\EDICE\ Dice}
%\index{dice}
%\idxboth{game-related}{symbols}
%\label{edice}
%\begin{tabular}{ll@{\qquad}ll@{\qquad}ll}
%  \KED[\allepsdice][\epsdice][\epsdice{1}]\epsdice\verb|{1}| &
%  \KED[\allepsdice][\epsdice][\epsdice{3}]\epsdice\verb|{3}| &
%  \KED[\allepsdice][\epsdice][\epsdice{5}]\epsdice\verb|{5}| \\
%  \KED[\allepsdice][\epsdice][\epsdice{2}]\epsdice\verb|{2}| &
%  \KED[\allepsdice][\epsdice][\epsdice{4}]\epsdice\verb|{4}| &
%  \KED[\allepsdice][\epsdice][\epsdice{6}]\epsdice\verb|{6}| \\
%\end{tabular}
%\end{symtable}
%\newif\ifEDICE
%\newcommand\EDICE{\pkgname{epsdice}}
%\IfFileExists{epsdice.sty}
%  {\EDICEtrue
%   \usepackage{epsdice}
%   \let\origepsdice=\epsdice
%   \DeclareRobustCommand{\epsdice}[1]{\origepsdice{##1}}
%   \DeclareRobustCommand{\allepsdice}{%
%     \epsdice{1}%
%     \epsdice{2}%
%     \epsdice{3}%
%     \epsdice{4}%
%     \epsdice{5}%
%     \epsdice{6}%
%   }
%  }
%  {}
```

The below are faked symbols, based on Scott Pakin's **fakedozenal.sty**. We include this file with the **phd** bundle distribution. With LuaLaTeX this can be included. This is a rather strange package with probably very few users and

use. It has intrigued me and I include it here.[8]

```
% Load a faked version of a package.
%\newcommand{\fakeusepackage}[1]{%
% \let\origProvidesPackage=\ProvidesPackage
% \def\ProvidesPackage##1[##2]{\origProvidesPackage{##1}[##2]\endinput}
% \usepackage{#1}
% \let\ProvidesPackage=\origProvidesPackage
% \usepackage{fake#1}
%}
%
%\newif\ifDOZ
%\newcommand\DOZ{\pkgname{dozenal}}
%\IfFileExists{dozenal.sty}
%  {\DOZtrue
%   \if\luatex
%      \RequirePackage[nocounters,typeone]{dozenal}
%    \else
%     \fakeusepackage{dozenal}
%   \fi
%   \DeclareRobustCommand{\DOZx}{\doz{X}}
%   \DeclareRobustCommand{\DOZe}{\doz{E}}
%   \DeclareRobustCommand{\alldoztallies}{%
%     \doz{1}~%
%     \doz{2}~%
%     \doz{3}~%
%     \doz{4}~%
%     \doz{5}~%
%     \doz{6}%
%   }
%  }
%{}
```

\KED[⟨⟩][⟨2⟩][⟨3⟩][⟨4⟩]

Documentation macros for tables showing dozenal symbols. Rarely used.

```
\def\KED[#1][#2][#3]#4 {%
   \incsyms\indexcommand[#1]{#2}#3 &\small\ttfamily\string#4%
}

% feyn provides yet another math font for which we have no room.
% Fortunately, it's relatively easy to define all of its symbols in
% terms of a text font.


\def\Kfeyn#1{\incsyms\indexcommand[\string\feyn{#1}]{\feyn{#1}}\feyn{#1} &\small\ttfamil
% We have no math alphabets left so we trick svrsymbols.sty into defining
% all of its characters in text mode.
%\newif\ifSVR
%\newcommand\SVR{\pkgname{svrsymbols}}
%\makeatletter
```

[8] See the website of the dozenal society to understand what it is all about. http://www.dozenal.org/index.html

```
%\IfFileExists{svrsymbols.sty}
%  {\SVRtrue
%   \let\origDeclareSymbolFont=\DeclareSymbolFont
%   \let\origDeclareMathSymbol=\DeclareMathSymbol
%   \let\origSetSymbolFont=\SetSymbolFont
%   \renewcommand{\DeclareSymbolFont}[5]{}
%   \renewcommand{\SetSymbolFont}[6]{}
%   \renewcommand{\DeclareMathSymbol}[4]{%
%     \let##1=\relax%  \photon and \antiproton are defined repeatedly.
%     \newcommand{##1}{{\usefont{OML}{svr}{m}{it}\char##4}}%
%   }
%   \usepackage{svrsymbols}
%   \let\DeclareSymbolFont=\origDeclareSymbolFont
%   \let\DeclareMathSymbol=\origDeclareMathSymbol
%   \let\SetSymbolFont=\origSetSymbolFont
%  }
%  {}
%%
%%
%  \newenvironment{longsymtable}[2][true]{%
%  \expandafter\global\expandafter\let
%    \expandafter\ifshowsymtable\csname if#1\endcsname
%  \ifshowsymtable
%    \mbox{}%
%    \Needspace*{13\baselineskip}%
%    \mbox{}%
%    \begin{center}%
%    \phantomsection
%    \refstepcounter{table}%
%%
%    % Inhibit longtable's implicit increment of the table counter.
%    \let\refstepcounter=\@gobble
%    \let\LT@array=\origLT@array
%    \let\LT@start=\origLT@start
%%
%    \addcontentsline{toc}{subsection}{%
%      \protect\numberline{\tablename~\thetable:}{#2}}%
%    \@makecaption{\fnum@table}{#2}%
%    \gdef\lt@indexed{}%
%    \let\next=\relax
%  \else
%    % The following was taken verbatim from verbatim.sty.
%    \let\do\@makeother\dospecials\catcode`\^^M\active
%    \let\verbatim@startline\relax
%    \let\verbatim@addtoline\@gobble
%    \let\verbatim@processline\relax
%    \let\verbatim@finish\relax
%    \let\next=\verbatim@
%  \fi
%  \next
%}{%
%  \ifshowsymtable
%    \end{center}
%    \let\@elt=\index\lt@indexed  % Close our index ranges.
```

```
%    \gdef\lt@indexed{}%
%    \vskip 8ex minus 2ex
%  \fi
%}
%\begin{longsymtable}[SVR]{\SVR\ Physics Ideograms}
%\ltindex{particle-physics symbols}
%\ltindex{symbols>particle physics}
%\ltindex{fermions}
%\ltindex{subatomic particles}
%\ltindex{photons}
%\label{svrsymbols}
%\begin{longtable}{*3{ll}}
%\multicolumn{6}{l}{\small\textit{(continued from previous page)}} \\[3ex]
%\endhead
%\endfirsthead
%\\[3ex]
%\multicolumn{6}{r}{\small\textit{(continued on next page)}}
%\endfoot
%\endlastfoot
%\K\adsorbate      & \K\experimentalsym & \K\protein        \\
%\K\adsorbent      & \K\externalsym     & \K\proton         \\
%\K\antimuon       & \K\fermiDistrib    & \K\quadrupole     \\
%\K\antineutrino   & \K\fermion         & \K\quark          \\
%\K\antineutron    & \K\Gluon           & \K\quarkb         \\
%\K\antiproton     & \K\graphene        & \K\quarkc         \\
%\K\antiquark      & \K\graviton        & \K\quarkd         \\
%\K\antiquarkb     & \K\hbond           & \K\quarks         \\
%\K\antiquarkc     & \K\Higgsboson      & \K\quarkt         \\
%\K\antiquarkd     & \K\hole            & \K\quarku         \\
%\K\antiquarks     & \K\interaction     & \K\reference      \\
%\K\antiquarkt     & \K\internalsym     & \K\resistivity    \\
%\K\antiquarku     & \K\ion             & \K\rhomesonminus  \\
%\K\anyon          & \K\ionicbond       & \K\rhomesonnull   \\
%\K\assumption     & \K\Jpsimeson       & \K\rhomesonplus   \\
%\K\atom           & \K\Kaonminus       & \K\solid          \\
%\K\bigassumption  & \K\Kaonnull        & \K\spin           \\
%\K\Bigassumption  & \K\Kaonplus        & \K\spindown       \\
%\K\biggassumption & \K\magnon          & \K\spinup         \\
%\K\Bmesonminus    & \K\maxwellDistrib  & \K\surface        \\
%\K\Bmesonnull     & \K\metalbond       & \K\svrexample     \\
%\K\Bmesonplus     & \K\method          & \K\svrphoton      \\
%\K\bond           & \K\muon            & \K\tachyon        \\
%\K\boseDistrib    & \K\neutrino        & \K\tauleptonminus \\
%\K\boson          & \K\neutron         & \K\tauleptonplus  \\
%\K\conductivity   & \K\nucleus         & \K\Tmesonminus    \\
%\K\covbond        & \K\orbit           & \K\Tmesonnull     \\
%\K\dipole         & \K\phimeson        & \K\Tmesonplus     \\
%\K\Dmesonminus    & \K\phimesonnull    & \K\triplecovbond  \\
%\K\Dmesonnull     & \K\phonon          & \K\Upsilonmeson   \\
%\K\Dmesonplus     & \K\pionminus       & \K\varphoton      \\
%\K\doublecovbond  & \K\pionnull        & \K\water          \\
%\K\electron       & \K\pionplus        & \K\Wboson         \\
%\K\errorsym       & \K\plasmon         & \K\Wbosonminus    \\
%\K\etameson       & \K\polariton       & \K\Wbosonplus     \\
```

```
%\K\etamesonprime  & \K\polaron         & \K\Zboson         \\
%\K\exciton        & \K\positron        &                   \\
%\end{longtable}
%\end{longsymtable}
```

```
%\def\Kpig#1{\incsyms\index{pigpenfont #1=\string\verb+{\string\pigpenfont\space#1}+\spa
%
%
%  \begin{docCommand} {Ks} { \marg{cmd} }
%    \cs{Ks} index and doc command, asterisk for note that is not available in |OT1|, as
%    superscript.
%  \end{docCommand}
%
%  \subsection{Arrows}
%%
% We use a faked version of old-arrows.sty provide by Pakin so as not to waste a math al
% using LuaLaTeX. Under the latest versions of Lua and LaTeX we load it directly.
%    \begin{macrocode}
%\newif\ifARR
%\newcommand\ARR{\pkgname{old-arrows}}
%\IfFileExists{old-arrows.sty}
%  {\ARRtrue\usepackage[old]{old-arrows}}
%  {}

%
%
%%\index{arrows}
%%\label{var-arrows}
%%\begin{tabular}{*3{ll}}
%%\X\vardownarrow\downarrow                    &  \X\varlongleftrightarrow \longleftright
%% \X\varhookleftarrow \hookleftarrow          &  \X\varlongmapsfrom \longmapsfrom$^*$
%% \X\varhookrightarrow \hookrightarrow        &  \X\varlongmapsto \longmapsto
%% \X\varleftarrow \leftarrow                  &  \X\varlongrightarrow \longrightarrow
%% \X\varleftrightarrow \leftrightarrow        &  \X\varmapsfrom \mapsfrom$^*$
%% \X\varlonghookrightarrow \longhookrightarrow &  \X\varmapsto \mapsto
%% \X\varlongleftarrow \longleftarrow          &  \X\varnearrow \nearrow
%%\end{tabular}

```

```
\ExplSyntaxOn
  \cs_set:Npn \Ks #1
    {
      \incsyms
      \indexcommand[\string\encone{\string#1}] {#1}
      { \encone{#1} }  & \ttfamily\string#1$^*$
    }
\ExplSyntaxOff
```

This macro is also from the comprehensive and takes the symbol command as its only argument. It provides **T1** encoding and also adds the command to the index.

```
\ExplSyntaxOn
\cs_set:Npn \Kt #1
  {
    \incsyms
```

```
    \indexcommand[\string\encone{\string#1}] {#1}
    {\encone{#1}} & \ttfamily \string #1
  }
\ExplSyntaxOff
```

\Kv {⟨*cmd*⟩}

T5 encoding

```
\def\Kv#1{\incsyms\indexcommand[\string\encfive{\string#1}]{#1}{\encfive{#1}} &\ttfamily
\def\Kgr@opt@arg[#1]#2{\incsyms\indexcommand[\string\encgreek{\string#1}]{#2}{\encgreek
  \def\Kgr@no@opt@arg#1{\incsyms\indexcommand[\string\encgreek{\string#1}]{#1}{\encgreek

\def\Kgr{\@ifnextchar[{\Kgr@opt@arg}{\Kgr@no@opt@arg}}
\def\KN[#1][#2]#3{\incsyms\indexcommand[\string#1]{#3} #1 & #2 & \ttfamily\string#3}
\def\KNbig[#1][#2]#3{\incsyms\indexcommand[\string#2]{#3} #1 & #2 & \ttfamily\string#3}

\def\Knoidx#1{\incsyms#1 &\ttfamily\string#1}
```

\N

Big delimiters auxiliary command for doc and index.

```
\ExplSyntaxOn
\cs_set:Npn \N@opt@arg #1 #2
  {
    \incsyms
    \indexcommand[$\string#1$]{#2}
    $#1$ & $\Big#1$ &\footnotesize\ttfamily\string#2
  }

\cs_set:Npn \N@no@opt@arg#1
  {
    \incsyms\indexcommand[$\string#1$]{#1}
    $#1$ & $\Big#1$ &\ttfamily\string#1
  }

\NewDocumentCommand {\N} { o m }
  {
    \IfNoValueTF {#1}
      { \N@no@opt@arg {#2}  }
      { \N@opt@arg {#1}{#2} }
  }

\ExplSyntaxOff
```

\Nn[⟨⟩]{⟨⟩}

```
  \def\Nn[#1]#2{%
    \incsyms\indexcommand[$\string\nathdouble\string#1$]{#2}%
    $\nathdouble#1$ & $\nathdouble{\Big#1}$ & \ttfamily\string#2%
 }

  \def\Nnt#1[#2]#3{%
    \incsyms\indexcommand{\triple}%
```

```
    $\nathtriple#2$ & $\nathtriple{\Big#2}$ &
    \ttfamily\expandafter\string\csname#1triple\endcsname\string#3}
  \def\Np@opt@args[#1]{\@ifnextchar[{\Np@two@opt@args[#1]}{\Np@one@opt@arg[#1]}}
  \def\Np@two@opt@args[#1][#2]#3{\incsyms\index{_=\string#2{} ($\string#1$)}$#1$ & $\Big
  \def\Np@one@opt@arg[#1]#2{\incsyms\indexpunct[$\string#1$]{#2}$#1$ & $\Big#1$ &\ttfam
  \def\Np@no@opt@args#1{\incsyms\indexpunct[$\string#1$]{#1}$#1$ & $\Big#1$ &\ttfamily\s
  \def\Np{\@ifnextchar[{\Np@opt@args}{\Np@no@opt@args}}
  \def\Nbig[#1]#2{\incsyms\indexcommand[$\string\Big\string#1$]{#2}$#1$ & $\Big#1$ &\tt
```

\Q

Used to typeset accents in a symbols table. Note it adds the (&). First we define auxiliary functions.

\Q@opt@arg

```
\ExplSyntaxOn
\cs_set:Npn \Q@opt@arg#1#2
  {
    \incsyms\indexaccent[\string\blackacchack{\string#1}]{#2}#1{A}#1{a} &
           \ttfamily\string#2\string{A\string}\string#2\string{a\string}
  }

\cs_set:Npn \Q@no@opt@arg#1
  {
    \incsyms\indexaccent[\protect\blackacchack{\string#1}]{#1}#1{A}#1{a} &
    \ttfamily\string#1\string{A\string}\string#1\string{a\string}
  }

\NewDocumentCommand {\Q} { o m }
  {
    \IfNoValueTF {#1}
      { \Q@no@opt@arg {#2} }
      { \Q@opt@arg    {#1}{#2} }
  }
\ExplSyntaxOff
```

(*End definition for* `\Q@opt@arg`.)

```
\newif\ifHARM
\newcommand\HARM{\pkgname{harmony}}
\IfFileExists{harmony.sty}
  {\HARMtrue
   \let\orignewcommand=\newcommand
   \let\newcommand=\DeclareRobustCommand
   %\savesymbol{HH}
   \usepackage{harmony}
   %\restoresymbol{harm}{HH}
   \let\newcommand=\orignewcommand
  }
  {}
```

Table C.4: **harmony**
Musical Accents

Āā	`\Ferli{A}\Ferli{a}`*	Ꙗa	`\Ohne{A}\Ohne{a}`*
Āā	`\Fermi{A}\Fermi{a}`	A a	`\Umd{A}\Umd{a}`*
Ⓐ ⓐ	`\Kr{A}\Kr{a}`		

`\Qc`{⟨*accent symbol command*⟩}

Typesets and indexes accents on two letters (A) and (a), as two tabular cells. Increases the `incsyms` counter.

```
\ExplSyntaxOn
\cs_set:Npn \Qc#1
  {
    \incsyms
    \indexaccent[\string\blackacc{\string#1}]{#1} #1 {A} #1{a}  &
    \ttfamily
         \string#1
         \token_to_str:N {
            A
         \token_to_str:N }
         \token_to_str:N #1
         \token_to_str:N {
           a
         \token_to_str:N }
  }
\ExplSyntaxOff
```

ānd \=

`\Qe`⟨*arg1*⟩⟨*arg2*⟩⟨*arg3*⟩

Limited use command to typeset and index `magic symbols`, that is symbols that might confuse MakeIndex and that they are escaped. It is easier to use `text commands`.

Example: `\Qe[\magicequal][\magicequalname]\=` produces Āā `\={A}\={a}`

```
\def\Qe[#1][#2]#3{%
  \incsyms
  \incsyms
  \index{_=\string#2{} (\string\blackacchack{\string#1})}%
  #3{A}#3{a} &
  \ttfamily\string#3\string{A\string}\string#3\string{a\string}}
\def\Qt#1{\incsyms\indexaccent[\string\encone{\string\blackacc{\string#1}}]{#1}{\encone
          \ttfamily\string#1\string{A\string}\string#1\string{a\string}}
\def\Qpc#1#2{\incsyms\indexcommand{#2}{\raisebox{1pt}{\tiny[#1]}} &
             \ttfamily\string#2\string{A\string}\string#2\string{a\string}}
```

```
\def\Qpfc[#1]#2{\incsyms\indexaccent[\string\encfour{\string\blackacchack{\string#1}}]{#
           \ttfamily\string#2\string{A\string}\string#2\string{a\string}}
\newif\ifFC\FCfalse
\ifFC
  \def\Qiv#1#2{\incsyms\indexaccent[\string\encfour{\string\blackacchack{\string#1}}]{#1
               \ttfamily\string#1\string{A\string}\string#1\string{a\string}$^#2$}

  \def\QivBAR#1{\incsyms\index{_=\string\magicVertname{}
                (\string\encfour{\string\blackacchack{\string\FCbar}})}
                \encfour{\FCbar{A}\FCbar{a}} &
                \ttfamily\string\|\string{A\string}\string\|\string{a\string}$^#1$}
\else
  \def\Qiv#1#2{\Qpc{T4}{#1}$^#2$}
  \def\QivBAR#1{\Qpc{T4}{\|}$^#1$}
\fi
\newif\ifVIET\VIETfalse
\ifVIET
  \def\Qv#1#2{\incsyms\indexaccent[\string\encfive{\string\blackacchack{\string#1}}]{#1
              \ttfamily\string#1\string{A\string}\string#1\string{a\string}$^#2$}
\else
  \def\Qv#1#2{\Qpc{T5}{#1}$^#2$}\def\Qv#1#2{Err}%TODO
\fi
```

\R [⟨*ams cmd*⟩] {⟨*cmd*⟩}

Used for variable size math operators

```
\ExplSyntaxOn
\cs_set:Npn \R@opt@arg#1#2
  {
    \incsyms
    \indexcommand[$\string#1$]{#2}
     $#1$ & $\displaystyle#1$ &\ttfamily\string#2
  }
 \cs_set:Npn \R@no@opt@arg#1
  {
    \incsyms
    \indexcommand[$\string#1$]{#1}
    $#1$ & $\displaystyle#1$ &\ttfamily\string#1
  }
\NewDocumentCommand {\R} { o m}
  {
    \IfNoValueTF {#1}
      { \R@no@opt@arg {#2}      }
      { \R@opt@arg    {#1} {#2} }
  }
\ExplSyntaxOff
%% T commands
```

Keep tall symbols from touching.

$\iint$ \iint $\iiint$ \iiint

$\iiiint$ \iiiint $\int\!\!\dots\!\int$ \idotsint

\indexDing {⟨*ding symbol number*⟩}

Auxiliary function to index and print in a table ding symbols. originally from Comprehensive.

```
\ExplSyntaxOn
\newcommand \indexDing [1]
  {
    \incsyms
    \indexcommand{\ding}
    \ding{#1} &
    \footnotesize\ttfamily\string\ding \string{#1\string}
  }
\ExplSyntaxOff
%    \end{macrococode}
%
%    \begin{macrocode}
\def\Tding#1{%
  \incsyms\indexcommand[\ding{#1}]{\ding{#1}}\ding{#1}\indexcommand{\ding} &
  \ttfamily\string\ding\string{#1\string}%
}

\def\Tm#1{\incsyms\indexcommand{\maya}$\mayadigit{#1}$ &\ttfamily\string\maya\string{#1\
\def\Tmoon#1{\incsyms\indexcommand{\MoonPha}\MoonPha{#1} &\ttfamily\string\MoonPha\strin
```

\indexTextcomp[⟨*ltx cmd*⟩] {⟨*symbol arg*⟩}

This command typesets its command argument in a table row of two (used for textcomp symbols).

```
\newcommand{\indexTextcomp}[2][]{%
   \incsyms#1 &
   \indexcommand[#2]{#2}% necessary to put symbol \text
   #2%
   &\ttfamily\string#2
}
```

\Vp

Commands that work both in math and text mode

```
\newcommand{\Vp}[2][]{\incsyms#1 & \indexpunct[$#2$]{#2}#2 &\ttfamily\string#2}
\newcommand{\V}[2][]{\incsyms\indexcommand[#1]{#2}#1 & \indexcommand[#2]{#2}#2 &\ttfamil
\newcommand{\Vl}[1]{\incsyms\indexcommand{#1}#1 & & \ttfamily\string#1}
\newcommand{\Vpl}[1]{\incsyms\indexpunct[$#1$]{#1}#1 & & \ttfamily\string#1}
\def\W@opt@arg[#1]#2#3{%
    \incsyms\indexaccent[$\string\blackacc{\string#1}$]{#2}%
    $#1{#3}$ &\ttfamily\string#2\string{#3\string}}

\def\W@no@opt@arg#1#2{%
    \incsyms\indexaccent[$\string\blackacc{\string#1}$]{#1}%
    $#1{#2}$ &\ttfamily\string#1\string{#2\string}}

\def\W{\@ifnextchar[{\W@opt@arg}{\W@no@opt@arg}}
```

```
\def\Wf#1#2{\incsyms\indexcommand{#1}$#1{#2}$ &\ttfamily\string#1\string{#2\string}}
\def\Ww#1#2#3{\incsyms\indexcommand{#2}$#1{#3}$ &\ttfamily\string#2\string{#3\string}}
\def\Wul#1#2#3{%
  \incsyms\indexaccent[$\string\blackacctwo{\string#1}$]{#1}%
  $#1{#2}{#3}$ &\ttfamily\string#1\string{#2\string}\string{#3\string}}
```

\X [⟨*command*⟩] {⟨*command*⟩}

Typesets its arguments as commands and also the resulting symbol in math. Used for symbol tables in the documentation.

Table C.6: AMS Commands Defined to Work in Both Math and Text Mode

✓	\checkmark	®	\circledR	✠	\maltese

U 25-6-2015

```
\ExplSyntaxOn
\def\X_opt_arg#1#2 {\incsyms\indexcommand[$\string#1$]{#2}$#1$ &\ttfamily\string#2}
\NewDocumentCommand{\X@no@opt@arg}{m}{\incsyms\indexcommand[$\string#1$]{#1}$#1$ &\ttfam

\NewDocumentCommand {\X} { o m}
  {
    \IfNoValueTF{#1}
      { \X@no@opt@arg  {#2}   }
      { \X_opt_arg {#1} {#2}  }
  }
\ExplSyntaxOff
% \def\X@opt@arg[#1]#2{\incsyms\indexcommand[$\string#1$]{#2}$#1$ &\ttfamily\string#2}
%  \def\X@no@opt@arg#1{\incsyms\indexcommand[$\string#1$]{#1}$#1$ &\ttfamily\string#1}
%  \def\X{\@ifnextchar[{\X@opt@arg}{\X@no@opt@arg}}
```

\Y{⟨*math command*⟩}

```
\def\Y#1{\incsyms\indexcommand[$\string\big\string#1$]{#1}$\big#1$ & $\Bigg#1$%
   &\scriptsize\ttfamily\string#1}
```

\docZ {⟨*arg1*⟩}

Typesets and index its arguments.

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

```
\ExplSyntaxOn
\cs_set:Npn \docZ #1
  {
    \incsyms
    \indexcommand[$\string#1$] {#1}
    \footnotesize
```

```
    \ttfamily
    \string #1
  }
\ExplSyntaxOff
```

\utfviii ⟨*void*⟩

Typesets UTF-8.

```
\NewDocumentCommand{\utfviii}{ }{\mbox{UTF-8}\index{UTF-8}\xspace}

% Index TeXbook symbols and the CTAN repository. CHECK
\newcommand{\idxTBsyms}{%
  \index{symbols>TeXbook=\TeX{}book}%
  \index{TeXbook, The=\TeX{}book, The>symbols from}%
}
```

\pkgname {⟨*package name*⟩}

Typesets and indexes a LaTeX package.

Lazarides

```
\newcommand{\pkgname}[1]{%
  \href{http://ctan.org/pkg/#1}{\bfseries{#1}}%
  \index{#1=\texttt{#1} (package)}%
  \index{packages>#1=\texttt{#1}}

  }
```

The \SpecialPackageIndex indexes a package using **hypdoc** style commands so that it can work both in doc as well as elsewhere.

```
\newcommand*\SpecialPackageIndex[2][packages:]{%
 \@bsphack
 \begingroup
 \HD@target
 \let\HDorg@encapchar\encapchar
 \edef\encapchar usage{%
 \HDorg@encapchar hdclindex{\the\c@HD@hypercount}{usage}%
 }%
 \index{#1\levelchar #2\actualchar
  {\protect\ttfamily #2}\encapchar usage}%
 \endgroup
 \@esphack
}

\newcommand{\pkg}[1]{%
  \href{http://ctan.org/pkg/#1}{\bfseries{#1}}%
  \SpecialPackageIndex[Packages:]{#1}}

% Comaptibility
\let\Lpack\pkgname

\newcommand*\opt[1]{\texttt{#1}}

```

```
\newcommand*\feat[1]{\texttt{#1}}


\newcommand{\optname}[2]{%
  \textsf{#2}%
  \index{#2=\textsf{#2} (\textsf{#1} package option)}%
  \index{package options>#2=\textsf{#2} (\textsf{#1})}}
```

\docClass{⟨*name of class*⟩}

Prints and indexes a LaTeX class.

```
\newcommand{\docClass}[1]{%
  \href{http://ctan.org/pkg/#1}{\bfseries{#1}}%
  \index{#1=\texttt{#1} (class)}%
  \index{classes>#1=\texttt{#1}}}

\let\Lpack\pkgname
```

This macro and all similar macros starting from doc typeset their argument and also add the argument to the index.

\docfilename {⟨*file name*⟩}

Typesets and indexes a file name.

```
\newcommand{\docfilename}[1]{%
  \texttt{#1}
  \index{#1\idx@actual{\phdindexprintcomc{#1}(file)}\encapchar hyperpage}}

\let\docFilename\docfilename
```

\docfileextension {⟨*file extension*⟩}

Typesets and indexes a file extension, such as \docfileextension{⟨*.tex*⟩} (.tex). You type the dot if you want it to appear in the index, which is a good idea.

```
\newcommand{\docfileextension}[1]{%
  \texttt{#1}%
  \index{#1\idx@actual{\texttt{#1} (file extension)}\encapchar hyperpage}
  \index{file extensions\idx@level #1\idx@actual{\texttt{#1}}\encapchar hyperpage}
}

\let\docFileExtension\docfileextension
\newcommand{\PSfont}[1]{%
  #1%
  \index{#1 (font)}%
  \index{fonts\index@level#1}%
}
\NewDocumentCommand{\person} { m m } {#1\index{#2, #1} #2}
```

\ctan{⟨*package name*⟩}

Provides a link to the ctan package repository

```
\DeclareRobustCommand\ctan[1]{%
  \textcolor{green}{%
      \href{http://www.ctan.org/pkg/#1} {{\bfseries #1}}%
  \footnote{\protect\url{http://www.ctan.org/pkg/#1}}}
  \index{Packages>#1}%
}
\newcommand{\idxCTAN}{%
  \index{Comprehensive TeX Archive Network=Comprehensive \string\TeX{} Archive Network}]
% Typeset a string in various encodings.
\newcommand{\encone}[1]{{\fontencoding{T1}\selectfont#1}}
\newcommand{\encfour}[1]{{\fontencoding{T4}\selectfont#1}}
\newcommand{\encfive}[1]{{\fontencoding{T5}\selectfont#1}}
\newcommand{\encgreek}[1]{{\fontencoding{LGR}\selectfont#1}}

% Various punctuation marks confuse makeindex when used directly.
\let\magicrbrack=]
\let\magicequal=\=
\DeclareRobustCommand{\magicequalname}{\texttt{\string\=}}
\DeclareRobustCommand{\magicvertname}{\texttt{|}}
\DeclareRobustCommand{\magicVertname}{\texttt{\string\|}}

% Vertically center a text-mode symbol.
\newsavebox{\tvcbox}
\newcommand*{\textvcenter}[1]{%
  \savebox{\tvcbox}{#1}%
  \raisebox{0.5\dp\tvcbox}{\raisebox{-0.5\ht\tvcbox}{\usebox{\tvcbox}}}%
}
% Many tables have notes beneath them.  Define an environment in which to
% display such a note, with an optional, superscripted math symbol
% preceding it.
\newenvironment{tablenote}[1][]{
  \makebox[1em]{\ensuremath{^{#1}}}%
  \begin{minipage}[t]{0.75\textwidth}%
  \setlength{\parskip}{2ex}
}{%
  \end{minipage}%
}

% Define various messages we reuse repeatedly.
\newcommand{\twosymbolmessage}{%
  \begin{tablenote}
    Where two symbols are present, the left one is the ``faked'' symbol
    that \latexe provides by default, and the right one is the ``true''
    symbol that \TC\ makes available.
  \end{tablenote}
}

\newcommand{\notpredefinedmessage}{%
  \begin{tablenote}[*]
    Not predefined in \latexe.  Use one of the packages
    \pkgname{latexsym}, \pkgname{amsfonts}, \pkgname{amssymb},
    \pkgname{txfonts}, \pkgname{pxfonts}, or \pkgname{wasysym}.
  \end{tablenote}
}
```

```

\newcommand{\notpredefinedmessageABX}{%
  \begin{tablenote}[*]
    Not predefined in \latexe.  Use one of the packages
    \pkgname{latexsym}, \pkgname{amsfonts}, \pkgname{amssymb},
    \pkgname{mathabx}, \pkgname{txfonts}, \pkgname{pxfonts}, or
    \pkgname{wasysym}.
  \end{tablenote}
}

\newcommand{\usetextmathmessage}[1][]{%
  \begin{tablenote}[#1]
    It's generally preferable to use the corresponding symbol from
    \vref{math-text} because the symbols in that table work
    properly in both text mode and math mode.
  \end{tablenote}
}



\newcommand{\usefontcmdmessage}[2]{%
  These symbols must appear either within the argument to \cmd{#1} or
  following the \cmd{#2} font-selection command within a scope%
}
% Define an environment in which to write a single table of symbols.  The
% environment looks a lot like a table, but it doesn't float, and it gets
% an entry in the table of contents as opposed to the list of tables.
%
% The first argument is a conditional.  The table will appear only if
% the value of the conditional is true.  The second argument is the
% table's caption.

\def\fnum@table{\tablename~\thetable}

\newenvironment{symtable}[2][true]{%
  \bgroup
  \expandafter\global\expandafter\let%
    \expandafter\ifshowsymtable\csname if#1\endcsname
  \ifshowsymtable
    \noindent%
    \begin{minipage}[t]{\linewidth}    % Prevent page breaks
    \begin{center}
    \refstepcounter{table}%
    \phantomsection
    \addcontentsline{toc}{subsection}{%
      \protect\numberline{\tablename~\thetable:}{#2}}%
    \@makecaption{\fnum@table}{#2}\medskip
    \let\next=\relax
  \else
    % The following was taken verbatim from verbatim.sty.
    \let\do\@makeother\dospecials\catcode`\^^M\active
    \let\verbatim@startline\relax
    \let\verbatim@addtoline\@gobble
    \let\verbatim@processline\relax
```

```
    \let\verbatim@finish\relax
    \let\next=\verbatim@
  \fi
  \next
}{%
  \ifshowsymtable
    \end{center}
    \end{minipage}
    \vskip 8ex minus 2ex
  \fi
  \egroup
}
```

We need a command that can typeset a symbol in the text and index it by its command name and also show the symbol in brackets next to it.

\docSymbol[⟨*optional explicit formatting*⟩]{⟨*command*⟩}

```
% Display and index a command, but not its symbol (\cmd).  \cmdI shows
% the symbol in the index, with optional explicit formatting.  \cmdX is
% the same as \cmdI, but with the optional argument hardwired to the
% command displayed in math mode.  \cmdW indexes an accent.  \cmdIp is
% also similar to \cmdI but formats its argument with \indexpunct
% instead of \indexcommand.
\ExplSyntaxOn
\newcommand{\docSymbol}[2][]{%
  \def\first@arg{#1}%
  \ifx\first@arg\@empty
    \texttt{\string#2} (#2)%
    \indexcommand[#2]{#2}%
  \else
    \texttt{\string#2} (#2)%
    \indexcommand[#1]{#2}%
  \fi
}
\ExplSyntaxOff
```

Table C.7: Fontawsome Currency Symbols

฿	\faBtc	₪	\faIls	₩	\faKrw	$	\faUsd
€	\faEur	₹	\faInr	₽	\faRub	₿	\faViacoin
£	\faGbp	¥	\faJpy	₺	\faTry		

Fontawesome defines \faBitcoin(฿) as a synonym for \faBtc(฿); \faCny(¥), \faYen(¥), and \faRmb(¥) as synonyms for faJpy(faJpy); faDollar(faDollar) as a synonym for faUsd(faUsd); faEuro(faEuro) as a synonym for faEur(faEur); faRouble(faRouble) and faRuble(faRuble) as synonyms for faRub(faRub); faRupee(faRupee) as a synonym for faInr(faInr); faShekel(faShekel) and faSheqel(faSheqel) as synonyms for faIls(faIls); faTurkishLira(faTurkishLira) as a synonym for faTry(faTry); and \faWon(₩) as a synonym for \faKrw(₩).

```
\newenvironment{nonsymtable}[1]{%
  \begin{table}[htbp]
  \centering
  \caption{#1}\medskip
}{%
  \end{table}
}



{
  \global\let\myempty=\@empty
  \global\let\mygobble=\@gobble
  \catcode`\@=12
  \gdef\getridofats#1@#2\relax{%
    \def\getridtest{#2}%
    \ifx\getridtest\myempty%
      \expandafter\def\expandafter\strippedat\expandafter{\strippedat#1}
    \else%
      \expandafter\def\expandafter\strippedat\expandafter{\strippedat#1\protect\printana
      \getridofats#2\relax%
    \fi%
  }

  \gdef\removeats#1{%
    \let\strippedat\myempty%
    \edef\strippedtext{\stripcommand#1}%
    \expandafter\getridofats\strippedtext @\relax%
  }

  \gdef\stripcommand#1{\expandafter\mygobble\string#1}
}


\def\printanat{\char`\@}

\def\declare{\afterassignment\pgfmanualdeclare\let\next=}
\def\pgfmanualdeclare{\ifx\next\bgroup\bgroup\color{red!75!black}\else{\color{red!75!bla


\let\textoken=\command
\let\endtextoken=\endcommand

\def\myprintocmmand#1{\texttt{\char`\\#1}}
```

\example{⟨*void*⟩}

A no parameter macro to typeset a one line example, in code.

Example: This is an example for β.

```
\def\example{\par\smallskip\noindent\textit{Example: }}
```

```
\def\themeauthor{\par\smallskip\noindent\textit{Theme author: }}


```

```
\def\indexoption#1{%
  \index{#1@\protect\texttt{#1} option}%
  \index{Graphic options and styles!#1@\protect\texttt{#1}}%
}

\def\itemcalendaroption#1{\item \declare{\texttt{#1}}%
  \index{#1@\protect\texttt{#1} date test}%
  \index{Date tests!#1@\protect\texttt{#1}}%
}
```

\begin{class}{⟨*class*⟩}
⟨*environment content*⟩
\end{class}

```
\def\class#1{%
  \list{}%
    {\leftmargin=2em\itemindent-\leftmargin\def\makelabel##1{\hss##1}}%
   \extractclass#1@\par\topsep=0pt
}
\def\endclass{\endlist}

\def\extractclass#1#2@{%
\item{{{\ttfamily\char`\\documentclass}#2{\ttfamily\char`\{\declare{#1}\char`\}}}}%
  \index{#1@\protect\texttt{#1} class}%
  \index{Classes!#1@\protect\texttt{#1}}}



\def\index@prologue{\section*{Index}\addcontentsline{toc}{section}{My Index}
  This index only contains automatically generated entries. A good
  index should also contain carefully selected keywords. This index is
  not a good index.
  \bigskip
}
\@ifundefined{c@IndexColumns}{\newcount\c@IndexColumns}{}
\c@IndexColumns=2
  \def\theindex{\@restonecoltrue
    \columnseprule \z@  \columnsep 29\p@
    \twocolumn[\index@prologue]%
       \parindent -30pt
       \columnsep 15pt
       \parskip 0pt plus 1pt
       \leftskip 30pt
       \rightskip 0pt plus 2cm
       \small
       \def\@idxitem{\par}%
    \let\item\@idxitem \ignorespaces}
  \def\endtheindex{\onecolumn}
\def\noindexing{\let\index=\@gobble}



\newcommand\symarrow[1]{%
  \index{#1\idx@actual\protect\texttt{#1} arrow tip}%
```

```
  \index{Arrow tips\idx@level#1\idx@actual\protect\texttt{#1}}%
  \texttt{#1}& yields thick
  \begin{tikzpicture}[arrows={#1-#1},thick,baseline]
    \useasboundingbox (0pt,-0.5ex) rectangle (1cm,2ex);
    \draw (0pt,.5ex) -- (1cm,.5ex);
  \end{tikzpicture} and thin
  \begin{tikzpicture}[arrows={#1-#1},thin,baseline]
    \useasboundingbox (0pt,-0.5ex) rectangle (1cm,2ex);
    \draw (0pt,.5ex) -- (1cm,.5ex);
  \end{tikzpicture}
}

\newcommand\sarrow[2]{
  \index{#1@\protect\texttt{#1} arrow tip}%
  \index{Arrow tips!#1@\protect\texttt{#1}}
  \index{#2@\protect\texttt{#2} arrow tip}%
  \index{Arrow tips!#2@\protect\texttt{#2}}
  \texttt{#1-#2}& yields thick
  \begin{tikzpicture}[arrows={#1-#2},thick,baseline]
    \useasboundingbox (0pt,-0.5ex) rectangle (1cm,2ex);
    \draw (0pt,.5ex) -- (1cm,.5ex);
  \end{tikzpicture} and thin
  \begin{tikzpicture}[arrows={#1-#2},thin,baseline]
    \useasboundingbox (0pt,-0.5ex) rectangle (1cm,2ex);
    \draw (0pt,.5ex) -- (1cm,.5ex);
  \end{tikzpicture}
}

\newcommand\carrow[1]{
  \index{#1@\protect\texttt{#1} arrow tip}%
  \index{Arrow tips!#1@\protect\texttt{#1}}
  \texttt{#1}& yields for line width 1ex
  \begin{tikzpicture}[arrows={#1-#1},line width=1ex,baseline]
    \useasboundingbox (0pt,-0.5ex) rectangle (1.5cm,2ex);
    \draw (0pt,.5ex) -- (1.5cm,.5ex);
  \end{tikzpicture}
}
```

\patternindex

```
%% Macros taken fro PGF manual
\newcommand\patternindex[1]{%
  \HD@target
  \let\HDorg@encapchar\encapchar
  \edef\encapchar usage{%
  \HDorg@encapchar hdclindex{\the\c@HD@hypercount}{usage}%
  }%
  \index{#1\idx@actual{\protect\texttt{#1} (pattern)}\encapchar usage}%
%  \index{#1\levelchar #2\actualchar
%  {\protect\ttfamily #2}\encapchar usage}%
  \index{Patterns\levelchar#1\actualchar{\protect\texttt{#1}}\encapchar usage}%
  \texttt{#1}&
  \begin{tikzpicture}[baseline=.5ex]
    % Background
    \pattern [path fading=west,pattern=checkerboard light gray]
```

```
      (0,0) rectangle (5cm,2em);
    \pattern [pattern=#1,pattern color=black] (0,0) rectangle +(1.5cm,2em);
    \pattern [pattern=#1,pattern color=blue] (1.75,0) rectangle +(1.5cm,2em);
    \pattern [pattern=#1,pattern color=red] (3.5,0) rectangle +(1.5cm,2em);
  \end{tikzpicture} \\[1ex]
}
```

bricks

(*End definition for* `\patternindex`.)

```
\newcommand\patternindexinherentlycolored[1]{%
  \HD@target
  \let\HDorg@encapchar\encapchar
  \edef\encapchar usage{%
    \HDorg@encapchar hdclindex{\the\c@HD@hypercount}{usage}%
  }%
  \index{#1\idx@actual{\protect\texttt{#1} pattern}\encapchar usage}%
  \index{Patterns\idx@level#1\idx@actual{\protect\texttt{#1}}\encapchar usage}%
  \arial #1 &
  \begin{tikzpicture}[baseline=.5ex]
    % Background
    \pattern [path fading=west,pattern=checkerboard light gray]
      (0,0) rectangle (5cm,2em);
    \pattern [pattern=#1,pattern color=blue] (0,0) rectangle +(5cm,2em);
  \end{tikzpicture} \\[1ex]
}


\newcommand\plotmarkentry[1]{%
  \index{#1@\protect\texttt{#1} plot mark}%
  \index{Plot marks!#1@\protect\texttt{#1}}
  \texttt{\char`\\pgfuseplotmark\char`\{\declare{#1}\char`\}} &
  \tikz\draw[color=black!25] plot[mark=#1,mark options={fill=examplefill,draw=black}] co
}
\newcommand\plotmarkentrytikz[1]{%
  \index{#1@\protect\texttt{#1} plot mark}%
  \index{Plot marks!#1@\protect\texttt{#1}}
  \texttt{mark=\declare{#1}} & \tikz\draw[color=black!25]
  plot[mark=#1,mark options={fill=examplefill,draw=black}]
    coordinates {(0,0) (.5,0.2) (1,0) (1.5,0.2)};\\
}



\ifx\scantokens\@undefined
  \PackageError{phd}{You need to use extended latex
    (elatex) or (pdfelatex) to process this document}{}
\fi

\begingroup
\catcode`|=0
\catcode`[= 1
\catcode`]=2
\catcode`\{=12
```

```
\catcode `\}=12
\catcode`\\=12 |gdef|find@example#1\end{codeexample}[|endofcodeexample[#1]]
|endgroup

\begingroup
\catcode`\^=7
\catcode`\^^M=13
\catcode`\ =13%
\gdef\returntospace{\catcode`\ =13\def {\space}\catcode`\^^M=13\def^^M{}}%
\endgroup

\begingroup
\catcode`\%=13
\catcode`\^^M=13
\gdef\commenthandler{\catcode`\%=13\def%{\@gobble@till@return}}
\gdef\@gobble@till@return#1^^M{}
\gdef\@gobble@till@return@ignore#1^^M{\ignorespaces}
\gdef\typesetcomment{\catcode`\%=13\def%{\@typeset@till@return}}
\gdef\@typeset@till@return#1^^M{{\def%{\char`\%}\textsl{\char`\%#1}}\par}
\endgroup

\define@key{codeexample}{width}{\setlength\codeexamplewidth{#1}}
\define@key{codeexample}{graphic}{\colorlet{thecodebackground}{#1}}
\define@key{codeexample}{code}{\colorlet{thecodebackground}{#1}}
\define@key{codeexample}{execute code}{\csname code@execute#1\endcsname}
\define@key{codeexample}{code only}[]{\code@executefalse}
\define@key{codeexample}{pre}{\def\code@pre{#1}}
\define@key{codeexample}{post}{\def\code@post{#1}}
\define@key{codeexample}{vbox}[]{\def\code@pre{\vbox\bgroup\setlength{\hsize}{\linewidth
6pt}}\def\code@post{\egroup}}
\define@key{codeexample}{ignorespaces}[]{\let\@gobble@till@return=\@gobble@till@return@
\define@key{codeexample}{leave comments}[]{\def\code@catcode@hook{\catcode`\%=12}\let\co
\def\code@pre{}
\def\code@post{}
\def\code@catcode@hook{}

\newdimen\codeexamplewidth
\newif\ifcode@execute
\newbox\codeexamplebox
\def\codeexample[#1]{%
  \begingroup%
  \code@executetrue
  \setlength\codeexamplewidth{4cm+7pt}
  \setkeys{codeexample}{#1}%
  \parindent0pt
  \begingroup%
  \par%
  \medskip%
  \let\do\@makeother%
  \dospecials%
  \obeylines%
  \@vobeyspaces%
  \catcode`\%=13%
  \catcode`\^^M=13%
```

```
  \code@catcode@hook%
  \relax%
  \find@example}
\def\endofcodeexample#1{%
  \endgroup%
  \ifcode@execute%
    \setbox\codeexamplebox=\hbox{%
      {%
        {%
          \returntospace%
          \commenthandler%
          \xdef\code@temp{#1}% removes returns and comments
        }%
        \colorbox{thecodebackground}{\color{black}\ignorespaces%
          \code@pre\expandafter\scantokens\expandafter{\code@temp\ignorespaces}\code@pos
      }%
    }%
    \ifdim\wd\codeexamplebox>\codeexamplewidth%
      \def\code@start{\par}%
      \def\code@flushstart{}\def\code@flushend{}%
      \def\code@mid{\parskip2pt\par\noindent}%
      \def\code@width{\linewidth-6pt}%
      \def\code@end{}%
    \else%
      \def\code@start{%
        \linewidth=\textwidth%
        \parshape \@ne 0pt \linewidth
        \leavevmode%
        \hbox\bgroup}%
      \def\code@flushstart{\hfill}%
      \def\code@flushend{\hbox{}}%
      \def\code@mid{\hskip6pt}%
      \def\code@width{\linewidth-12pt-\codeexamplewidth}%
      \def\code@end{\egroup}%
    \fi%
    \code@start%
    \noindent%
    \begin{minipage}[t]{\codeexamplewidth}\raggedright
      \hrule width0pt%
      \small%\vskip-1em%
      \code@flushstart\box\codeexamplebox\code@flushend%
      \vskip-1ex
      \leavevmode%
    \end{minipage}%
  \else%
    \def\code@mid{\par}
    \def\code@width{\linewidth-6pt}
    \def\code@end{}
  \fi%
  \code@mid%
  \colorbox{thecodebackground}{%
    \begin{minipage}[t]{\code@width}%
      {%
        \let\do\@makeother
```

```
        \dospecials
        \frenchspacing\@vobeyspaces
        \normalfont\ttfamily%\footnotesize
        \typesetcomment%
        \@tempswafalse
        \def\par{%
          \if@tempswa
          \leavevmode \null \@@par\penalty\interlinepenalty
          \else
          \@tempswatrue
          \ifhmode\@@par\penalty\interlinepenalty\fi
          \fi}%
        \obeylines
        \everypar \expandafter{\the\everypar \unpenalty}%
        #1}
    \end{minipage}}%
  \code@end%
  \par%
  \medskip
  \end{codeexample}
}

\def\endcodeexample{\endgroup}
```

From pgfplots manual

```
\long\def\codeexamplenl{\noexpand\par}%
\pgfqkeys{/codeexample}{%
every codeexample/.style={
width=3.9cm,
/pgfplots/every axis/.append style={legend style={fill=thecodebackground}}
},
narrow/.style={width=6.9cm},
%tabsize=4,
%pre={\begin{minipage}{\linewidth}\begingroup},
%post={\endgroup\end{minipage}},
%vbox,
%newline=\codeexamplenl,
}
```

\keyval⟨*key*⟩⟨*options*⟩⟨*text*⟩

The macro **\keyval** typesets, key value lists and their options.
\keyval{test}{\marg{option1|option2|option2|option4}{text}

test = {⟨*option1|option2|option2|option4*⟩} Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

We first measure the width of the option and not use it (want to make it a bit flexible at a later stage. We also ensure that the catcode of **|** is set properly in case anyone is using short verbatim commands, as we do in this document.

```
\newlength\temp@cx
\def\keyval{%
```

```
  \begingroup
  \catcode`|=11
  \@keyval}
%
% use a list not very good
\def\@keyval#1#2#3{%
  \settowidth\temp@cx{\verbatimfont\bfseries{#1\thinspace}}%
  \hangindent\temp@cx
  \par\leavevmode%
\hspace*{-\temp@cx}{\verbatimfont\color{theoption}\bfseries #1}%
          \thinspace=\thinspace#2%
\hspace*{.5em}#3%
\par\addvspace{6.5pt}%
\endgroup
}
%
```

Typesets a sample of bib

```
\newenvironment{bibsample}
  {\trivlist\samepage
   \setlength{\itemsep}{0pt}}
  {\endtrivlist}
%% doccommands
\newcommand*{\marglistfont}{\itshape}

\newcommand*{\margoptionfont}{\ttfamily}

\newcommand*{\margnotefont}{}

\newcommand*{\optionlistfont}{\bfseries}

\newcommand*{\ltxsyntaxfont}{\ttfamily}

\newcommand*{\ltxsyntaxlabelfont}{\bfseries}

\newcommand*{\changelogfont}{\normalfont}

\newcommand*{\changeloglabelfont}{\bfseries}



%\def\cmd#1{\cs{\expandafter\cmd@to@cs\string#1}}%

%\def\cmd@to@cs#1#2{\char\number`#2\relax}

\newrobustcmd*{\env}[1]{\mbox{\verbatimfont\bfseries\textcolor{thegreen}{#1}}}

\newrobustcmd*{\len}[1]{\mbox{\verbatimfont\textbackslash#1}}

\newrobustcmd*{\cnt}[1]{\mbox{\verbatimfont#1}}

\newlength{\marglistsep}

\newlength{\marglistwidth}
```

```

%%% POSSIBLE ERROR
\setlength{\marglistwidth}{\oddsidemargin}%

\deflength{\marglistsep}{10pt}
%% This needs thorough checking as to restore previous definitions
%% of parsep we want parsep to be a bit higher than standard enumerated lists.
\global\newlength\varparsep
\newenvironment*{marglist}
  {\setlength\varparsep{\parsep}\list{}{%
     \parsep 3.5\p@ \@plus0\p@ \@minus\p@
     \setlength{\labelwidth}{\marglistwidth}%
     \setlength{\labelsep}{\marglistsep}%
     \setlength{\leftmargin}{0pt}%
     \renewcommand*{\makelabel}[1]{\hss\marglistfont##1}}}
  {\endlist\setlength\parsep{\varparsep}}

%
\newenvironment*{margoptionslist}
  {\setlength\varparsep{\parsep}\list{}{%
     \parsep 3.5\p@ \@plus0\p@ \@minus\p@
     \setlength{\labelwidth}{\marglistwidth}%
     \setlength{\labelsep}{\marglistsep}%
     \setlength{\leftmargin}{0pt}%
     \renewcommand*{\makelabel}[1]{\hss\margoptionfont\detokenize{##1}}}}
  {\endlist\setlength\parsep{\varparsep}}


```

\begin{keymarglist} ⟨*void*⟩
⟨*environment content*⟩
\end{keymarglist}

Typesets a key options list in the margin.

```
\newenvironment*{keymarglist}
  {\marglist
   \setlength{\itemsep}{0pt}%
   \raggedright}
  {\endmarglist}
% color definitions
\def\cvaref#1{\verbatimfont\textcolor{themacro}{#1}}
% color for options
\def\colOpt#1{\textcolor{theoption}{\verbatimfont\texttt{#1}}}
```

\option{⟨*option*⟩}

Typesets an option. It uses the color `theoption`, which is defined in the package **phd-colorpalette**.

```
\newcommand{\option}[1]{\colOpt{#1}}
```

C.16.3 Creating a Small Verbatim Environment

This is a modified version from Cambridge classes HAS ERRORS

```
\begingroup \catcode `|=0
\catcode `[= 1
\catcode`]=2
\catcode `\{=12
\catcode `\}=12
\catcode`\\=12
|gdef|@xsmallverbatim#1\end{smallverbatim}[#1|end[smallverbatim]]
|gdef|@sxsmallverbatim#1\end{smallverbatim*}[#1|end[smallverbatim*]]
|endgroup
\def\@smallverbatim{\trivlist \item\relax
  \if@minipage\else\vskip\parskip\fi
  \leftskip\@totalleftmargin\rightskip\z@skip
  \parindent\z@\parfillskip\@flushglue\parskip\z@skip
  \@@@@par %chaned
  \@tempswafalse
  \def\par{%
    \if@tempswa
      \leavevmode \null \@@par\penalty\interlinepenalty
    \else
      \@tempswatrue
      \ifhmode\@@@@par\penalty\interlinepenalty\fi
    \fi}%
  \let\do\@makeother \dospecials
  \obeylines \small \@noligs%\smallverbatim@font to FIX
  \hyphenchar\font\m@ne
  \everypar \expandafter{\the\everypar \unpenalty}%
}
\def\smallverbatim{\@smallverbatim \frenchspacing\@vobeyspaces \@xsmallverbatim}
\def\endsmallverbatim{\if@newlist \leavevmode\fi\endtrivlist}
\def\smallverbatim@font{\normalfont\smallverbatimsize\ttfamily}
```

This is a short test. Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

```
%  \ifx\bhj
%  \else
%  \fi
%
```

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

```
\begin{docCommands}
 ⟨environment content⟩
\end{docCommands}
```

```
\let\luacmd\docValue
\newenvironment{docCommands}{%
\bgroup
\par
\parindent=0pt
```

```
\parskip=3.5pt plus0.5pt
\everypar{\hangindent2em}%
\addvspace\belowdisplayskip\relax}%
{\everypar{}%
 \par
 \vskip\belowdisplayskip\egroup\par}
\long\def\auxm#1(#2);{%
  \def\Xtemp{#1}%
  \def\Ytemp{#2}%
  \parindent=0pt
  \addvspace{1.5pt}%
  \par\leavevmode
  \hangafter=1\relax   \hangindent=1em\relax
  \bgroup
   \bfseries\sffamily\color{red}\Xtemp\,\color{black}(\textit{\Ytemp})\hskip0.1em
  \egroup
}


\newenvironment{docLua}[1]{%
  \auxm#1;
 }{%
\@@par
\smallskip\parindent=1em }


```

`\docFont`{⟨*font name*⟩}

Typeset and indexes a font by name, such as `\docFontArial` typesetting "Arial" .

```
    \DeclareRobustCommand{\phdidxfont}[1]{%
      \index{#1 (font)}\index{fonts\idx@level{#1}\encapchar hyperpage}}%
    \def\docFont#1{
        \enquote{#1}
        \phdidxfont{#1}%
    }
```

Key handler ⟨*key*⟩/`.fontweight`

A handler to handle fontweights. Chooses betwenn.This handler causes the default path to be set to hkeyi. Note that the default path is reset at the beginning of each call to pgfkeys to be equal to ...TODO remove second parameter as it is not needed. Add index command

```
\newenvironment{handler}[2][]{%
  \begin{phdd_manual_entry}%
   \begingroup
   \sffamily\textbf{Key handler} \meta{key}/\bfseries\ttfamily{\color{thered}#2}\color{b
   \endgroup
 \end{phdd_manual_entry}
 }
{}
```

```
\newcommand\stok[2][]{%
  \texttt{#2}\ensuremath{_{%
```

```
  \if!#1!
    \the\catcode`#2
  \else
    #1
  \fi}}}
```

C.17 Unicode math index functions

The functions that follow typeset unicode math tables.

\showsymbolalpha {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

Indexes and typesets all the alphabetic letters available in math, mostly greek and the dotless j and i.

```
\newcommand\showsymbolalpha[3]
  {
    \par\noindent\hangindent=3em%
    \makebox[2em][l]{$#1$} \makebox[3.5em][l]{\texttt{U+#2}}
    \string#1 $^{#3}$
    \indexmathcmd [Math alphabetics] {#1}
  }
```

\showsymbol {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

```
\newcommand\showsymbol[3]{\par\noindent\hangindent=3em%
\makebox[2em][l]{$#1$} \makebox[3.5em][l]{\texttt{U+#2}}
\string#1 $^{#3}$\indexmathcmd [Math ordinary] {#1}}
```

\showsymbolbin {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

```
\newcommand\showsymbolbin[3]{\par\noindent\hangindent=3em%
\makebox[2em][l]{$#1$} \makebox[3.5em][l]{\texttt{U+#2}}
\string#1 $^{#3}$\indexmathcmd [Math bin operators] {#1}}
```

\showrelsymbol {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

```
\newcommand\showrelsymbol[3]{\par\noindent\hangindent=3em%
\makebox[2em][l]{$#1$} \makebox[3.5em][l]{\texttt{U+#2}}
\string#1$^{#3}$\indexmathcmd [Math relations] {#1}}
```

\integralsymbol {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

Typesets and inserts into index integral symbols

```
\newcommand\integralsymbol[3]{\par\noindent\hangindent=3em%
\makebox[2em][l]{$#1$} \makebox[3.5em][l]{\texttt{U+#2}}
\string#1 $^{#3}$\indexmathcmd [Math integrals] {#1}}
```

\showop {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

Typesets and inserts into index integral symbols

```
\newcommand\showop[3]{\par\noindent\hangindent=6em%
  \makebox[5em][l]{$#1$\hfill$\displaystyle#1$\hfill}
  \makebox[3.5em][l]{\small\texttt{U+#2}} \string#1 $^{#3}$
  \indexmathcmd [Math big operators] {#1} }
```

\showmbrace {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

Typesets and inserts middle brace symbols

```
\newcommand\showmbrace[3]{\par\noindent\hangindent=6em%
  \makebox[5em][l]{${#1}{\bigm#1}{\Bigm#1}{\biggm#1}{\Biggm#1}$}
  \makebox[3.5em][l]{\small\texttt{U+#2}} \string#1 $^{#3}$
  \indexmathcmd [Math delimiters]{#1}
  }
```

\showlbrace {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

left braces

```
\newcommand\showlbrace[3]{\par\noindent\hangindent=6em%
  \makebox[5em][l]{$\Biggl#1\biggl#1\Bigl#1\bigl#1#1$}
  \makebox[3.5em][l]{
    \small\texttt{U+#2}} \string #1 $^{#3}$
  \indexmathcmd[Math delimiters] {#1}
  }
```

\showrbrace {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

right braces

```
\newcommand\showrbrace[3]{\par\noindent\hangindent=6em%
  \makebox[5em][l]{$#1\bigr#1\Bigr#1\biggr#1\Biggr#1$}
  \makebox[3.5em][l]{\small\texttt{U+#2}} \string #1 $^{#3}$
  \indexmathcmd[Math delimiters] {#1}
  }
```

\wide accents {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

wide accents

```
\DeclareDocumentCommand \showwideaccent { m m m} {\par\noindent\hangindent=4em%
  \makebox[3em][l]{$#1{xxx}$}\makebox[3.5em][l]{\small\texttt{U+#2}} \string #1 $^{#3}$
  \indexmathcmd [Math accents] {#1{abc}}
  }
```

\showaccent {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

right braces

```
\DeclareDocumentCommand\showaccent { m m m} {\par\noindent\hangindent=4em%
  \makebox[3em][l]{$#1b$}\makebox[3.5em][l]{\small\texttt{U+#2}} \string #1$^{#3}$
  \indexmathcmd [Math accents] {#1 b}
  }
```

\showrover {⟨*cmd*⟩} {⟨*unicode point*⟩} {⟨*note symbol*⟩}

```
\newcommand\showover[3]{\par\noindent\hangindent=6em%
  \makebox[5em][l]{$#1{xxxxxx}$}
  \makebox[3.5em][l]{\small\texttt{U+#2}}
  \string #1 $^{#3}$
  \indexmathcmd [Math over and under brackets] {#1{xxxxxx}}
  }
\ExplSyntaxOn
\NewDocumentEnvironment{ arguments }  {}
  {
    \enumerate
  }
  {
    \endenumerate
  }
\ExplSyntaxOff
```

l3 Documentation

The package provides macros for documenting and indexing primarily LaTeX or LaTeX 3 code. It combines the styles originitating with `doc` with those found in the `pgf` series of packages. It removes some of the difficulties of using hyperlinks in the index and elsewhere in the document. 21.00002

16

-2.5

-2.1

0.87865

The LaTeX 3 Team developed the class **l3doc** as a replacement to the **ltxdoc** class. This is currently being used to document the **expl3** and related packages.

This section of the package deals primarily with an adaptation from the class.

The main reason it is easier to integrate with the rest of the **phd** bundle if the code is in a package, as in the original **doc** and **ltxdoc**.

D.0.1 General text markup

Many of the commands in this section come from **ltxdoc** with some improvements.

\cmd **\cmd** [⟨*options*⟩] ⟨*control sequence*⟩
\cs **\cs** [⟨*options*⟩] {⟨*csname*⟩}

These commands are provided to typeset control sequences. **\cmd\foo** produces "**\foo**" and **\cs{foo}** produces the same. In general, **\cs** is more robust since it doesn't rely on catcodes being "correct" and is therefore recommended.

These commands are aware of the **@@** **l3docstrip** syntax and replace such instances correctly in the typeset documentation. This only happens after a %<@@=⟨*module*⟩> declaration.

Additionally, commands can be used in the argument of **\cs**. For instance, **\cs{\meta{name}:\meta{signature}}** produces \⟨*name*⟩:⟨*signature*⟩.

The ⟨*options*⟩ are a key–value list which can contain the following keys:

- **index=**⟨*name*⟩: the ⟨*csname*⟩ is indexed as if one had written **\cs**{⟨*name*⟩}.
- **no-index**: the ⟨*csname*⟩ is not indexed.
- **module=**⟨*module*⟩: the ⟨*csname*⟩ is indexed in the list of commands from the ⟨*module*⟩; the ⟨*module*⟩ can in particular be **TeX** for "TeX and LaTeX 2_ε" commands, or empty for commands which should be placed in the main index. By default, the ⟨*module*⟩ is deduced automatically from the command name.
- **replace** is a boolean key (true by default) which indicates whether to replace **@@** as **l3docstrip** does.

These commands allow hyphenation of control sequences after (most) underscores. By default, a hyphen is used to mark the hyphenation, but this can be changed with the cs-break-nohyphen class option. To disable hyphenation of control sequencies entirely, use cs-break-off.

\tn **\tn** [⟨*options*⟩] {⟨*csname*⟩}

Analoguous to **\cs** but intended for "traditional" TeX or LaTeX 2_ε commands; they are indexed accordingly. This is in fact equivalent to **\cs [module=TeX, replace=false,** ⟨*options*⟩] {⟨*csname*⟩}.

\meta `\meta {⟨name⟩}`

`\meta` typesets the ⟨*name*⟩ italicised in ⟨*angle brackets*⟩. Within a `function` environment or similar, angle brackets `<...>` are set up to be a shorthand for `\meta{...}`.

This function has additional functionality over its **ltxdoc** versions; underscores can be used to subscript material as in math mode. For example, `\meta{arg_{xy}}` produces "⟨arg_{xy}⟩".

Function environments take an optional argument to indicate whether the function(s) it describes are expandable or restricted-expandable or defined in conditional forms. Use `EXP`, `rEXP`, `TF`, `pTF`, or `noTF` for this; note that `pTF` implies `EXP` since predicates must always be expandable, and that `noTF` means that the function without `TF` should be documented in addition to `TF`. As an example:

Example 0.1 Typesetting functions

```
% \begin{function}[pTF]{\cs_if_exist:N}
%   \begin{syntax}
%     \cs{cs_if_exist_p:N} \meta{cs}
%   \end{syntax}
% \end{function}
%
```

\cs_if_exist_p:N ⋆ **\cs_if_exist:NTF** ⋆ `\cs_if_exist_p:N ⟨cs⟩`

This function has additional functionality over its **ltxdoc** versions; underscores can be used to subscript material as in math mode. For example, `\meta{arg_{xy}}` produces "⟨arg_{xy}⟩".

\Arg **\marg** **\oarg** **\parg** `\Arg {⟨name⟩}` `\marg {⟨name⟩}`

Typesets the ⟨*name*⟩ as for `\meta` and wraps it in braces.

The `\marg`/`\oarg`/`\parg` versions follow from **ltxdoc** in being used for "mandatory" or "optional" or "picture" brackets as per LaTeX 2_ε syntax.

\file **\env** **\pkg** **\cls** `\pkg {⟨name⟩}`

These all take one argument and are intended to be used as semantic commands for representing files, environments, package names, and class names, respectively.

D.0.2 Describing functions in the documentation

`function` `syntax` Two heavily-used environments are defined to describe the syntax of **expl3** functions and variables.

```
% \begin{function}{\function_one:, \function_two:}
%   \begin{syntax}
%     |\foo_bar:| \Arg{meta} \meta{test_1}
%   \end{syntax}
% \meta{description}
% \end{function}
%
```

`\function_one:` `\foo_bar:` {⟨*meta*⟩} ⟨*test*$_1$⟩

`\function_two:` ⟨*description*⟩

Function environments take an optional argument to indicate whether the function(s) it describes are expandable or restricted-expandable or defined in conditional forms. Use `EXP`, `rEXP`, `TF`, `pTF`, or `noTF` for this; note that `pTF` implies `EXP` since predicates must always be expandable, and that `noTF` means that the function without `TF` should be documented in addition to `TF`. As an example:

```
% \begin{function}[pTF]{\cs_if_exist:N}
%   \begin{syntax}
%     \cs{cs_if_exist_p:N} \meta{cs}
%   \end{syntax}
% \meta{description}
% \end{function}
```

This code will then produce the output, as shown below and place it in the margin.

`\cs_if_exist_p:N` ⋆ `\cs_if_exist_p:N` ⟨*cs*⟩

`\cs_if_exist:NTF` ⋆

⟨*description*⟩

`variable` If you are documenting a variable instead of a function, use the `variable` environment instead; it behaves identically to the `function` environment above.

`texnote` This environment is used to call out sections within `function` and similar that are only of interest to seasoned TEX developers.

D.1 Describing functions in the implementation

`macro` The well-used environment from LATEX 2$_\varepsilon$ for marking up the implementation of

macros/functions remains the `macro` environment. Some changes in **l3doc**: it now accepts comma-separated lists of functions, to avoid a very large number of consecutive `\end{macro}` statements. Spaces and new lines are ignored (the option `[verb]` prevents this).

```
%  \begin{macro}{\foo:N, \foo:c}
%    \begin{macrocode}
% ... code for \foo:N and \foo:c ...
%    \end{macrocode}
%  \end{macro}
```

If you are documenting an auxiliary macro, it's generally not necessary to highlight it as much and you also don't need to check it for, say, having a test function and having a documentation chunk earlier in a `function` environment. **l3doc** will pick up these cases form the presence of `__` in the name, or you may force marking as internal by using `\begin{macro}[int]` to mark it as such. The margin call-out is then printed in grey for such cases.

For documenting **expl3**-type conditionals, you may also pass this environment a `TF` option (and omit it from the function name) to denote that the function is provided with `T`, `F`, and `TF` suffixes. A similar `pTF` option prints both `TF` and `_p` predicate forms. An option `noTF` prints both the `TF` forms and a form with neither `T` nor `F`, to document functions such as `\prop_get:NN` which also have conditional forms (`\prop_get:NNTF`).

`\TestFiles` `\TestFiles`{⟨*list of files*⟩} is used to indicate which test files are used for the current code; they are printed in the documentation.

`\UnitTested` Within a `macro` environment, it is a good idea to mark whether a unit test has been created for the commands it defines. This is indicated by writing `\UnitTested` anywhere within `\begin{macro}` ...`\end{macro}`.

If the class option `checktest` is enabled, then it is an *error* to have a `macro` environment without a call to `Testfiles`. This is intended for large packages such as **expl3** that should have absolutely comprehensive tests suites and whose authors may not always be as sharp at adding new tests with new code as they should be.

`\TestMissing` If a function is missing a test, this may be flagged by writing (as many times as needed) `\TestMissing`{⟨*explanation of test required*⟩}. These missing tests are summarised in the listing printed at the end of the compilation run.

`variable` When documenting variable definitions, use the `variable` environment instead. Here it behaves identically to the `macro` environment, except that if the class option `checktest` is enabled, variables are not required to have a test file.

`arguments` Within a `macro` environment, you may use the `arguments` environment to describe the arguments taken by the function(s). It behaves like a modified enumerate environment.

```
%  \begin{macro}{\foo:nn, \foo:VV}
%  \begin{arguments}
%    \item Name of froozle to be frazzled
%    \item Name of muble to be jubled
%  \end{arguments}
%    \begin{macrocode}
% ... code for \foo:nn and \foo:VV ...
```

```
%    \end{macrocode}
%  \end{macro}
```

```
\ExplSyntaxOn
```

```
%\RequirePackage{fancybox}
%\newif\iffancyvrb \IfFileExists{fancyvrb.sty}{\fancyvrbtrue}{}
%\iffancyvrb
%  \RequirePackage{fancyvrb}
%\fi
%\fvset{gobble=2,frame=none, framesep=5pt,fillcolor=red!30,numbers=left,
%          numbersep=3pt,fontsize=\small}
```

D.2 Variables

`\g_docinput_clist` The list of files which have been input through `\DocInput`.

```
@@=phdd
```

```
\clist_new:N \g_docinput_clist
```

All functions documented through `function`, and all macros introduced through `macro`. They can be compared to see what documentation or code is missing.

```
\seq_new:N \g_doc_functions_seq
\seq_new:N \g_doc_macros_seq
```

`\l__phdd_detect_internals_bool`

`\l__phdd_detect_internals_tl`

If `true`, **l3doc** will check for use of internal commands `\cs[no-index]{__\meta{pkg}_\ldots{}}` from other packages in the argument of the `macro` environment, and in the code typeset in `macrocode` environments, but not in `\cs`. Also a token list to store temporary data for this purpose.

```
\bool_new:N \l__phdd_detect_internals_bool
\bool_set_true:N \l__phdd_detect_internals_bool
\tl_new:N \l__phdd_detect_internals_tl
\tl_new:N \l__phdd_detect_internals_cs_tl
```

`\l__phdd_output_coffin`

The `function` environment is typeset by combining coffins containing various pieces (function names, description, *etc.*) into this coffin.

```
\coffin_new:N \l__phdd_output_coffin
```

`\l__phdd_functions_coffin`

`\l__phdd_descr_coffin`

`\l__phdd_syntax_coffin`

These coffins contain respectively the list of function names (argument of the `function` environment), the text between **`\begin{function}`** and **`\end{function}`**, and the syntax given in the `syntax` environment.

```
\coffin_new:N \l__phdd_functions_coffin
\coffin_new:N \l__phdd_descr_coffin
\coffin_new:N \l__phdd_syntax_coffin
```

`\g__phdd_syntax_box`

The contents of the `syntax` environment are typeset in this box before being transferred to **`\l__phdd_syntax_coffin`**.

```
\box_new:N \g__phdd_syntax_box
```

`\l__phdd_in_function_bool`

True when inside a `function` or `variable` environment. Used by the `syntax` environment to determine its behaviour.

```
\bool_new:N \l__phdd_in_function_bool
```

`\l__phdd_long_name_bool`

`\l__phdd_trial_width_dim`

The boolean **`\l__phdd_long_name_bool`** is `true` if the width **`\l__phdd_trial_width_dim`** of the coffin **`\l__phdd_functions_coffin`** (containing the current function names) is bigger than the space available in the margin.

```
\bool_new:N \l__phdd_long_name_bool
\dim_new:N \l__phdd_trial_width_dim
```

`\l__phdd_nested_macro_int`

The nesting of `macro` environments (this is now 0 outside a `macro` environment).

```
\int_new:N \l__phdd_nested_macro_int
```

\l__phdd_macro_tested_bool

\g__phdd_missing_tests_prop

\g__phdd_not_tested_seq

\g__phdd_testfiles_seq

A boolean describing whether the current macro has tests, and some global structures which contain information about test files and which tests are missing.

```
\bool_new:N \l__phdd_macro_tested_bool
\prop_new:N \g__phdd_missing_tests_prop
\seq_new:N \g__phdd_not_tested_seq
\seq_new:N \g__phdd_testfiles_seq
```

\l__phdd_macro_internal_set_bool

\l__phdd_macro_internal_bool

\l__phdd_macro_TF_bool

\l__phdd_macro_pTF_bool

\l__phdd_macro_noTF_bool

\l__phdd_macro_EXP_bool

\l__phdd_macro_rEXP_bool

\l__phdd_macro_var_bool

\l__phdd_override_module_tl

\l__phdd_macro_documented_tl

Contain information about some options of function/macro environments. We initialize **\l__phdd_override_module_tl** to avoid overriding module names by an empty name (meaning no module).

```
\bool_new:N \l__phdd_macro_internal_set_bool
\bool_new:N \l__phdd_macro_internal_bool
\bool_new:N \l__phdd_macro_TF_bool
\bool_new:N \l__phdd_macro_pTF_bool
\bool_new:N \l__phdd_macro_noTF_bool
\bool_new:N \l__phdd_macro_EXP_bool
\bool_new:N \l__phdd_macro_rEXP_bool
\bool_new:N \l__phdd_macro_var_bool
\tl_new:N \l__phdd_override_module_tl
\tl_set:Nn \l__phdd_override_module_tl { \q_no_value }
\tl_new:N \l__phdd_macro_documented_tl
```

\g__phdd_lmodern_bool

\g__phdd_checkfunc_bool

\g__phdd_checktest_bool

\g__phdd_cs_break_bool

\g__phdd_kernel_bool

Information about package options.

```
\bool_new:N \g__phdd_lmodern_bool
\bool_new:N \g__phdd_checkfunc_bool
\bool_new:N \g__phdd_checktest_bool
\bool_new:N \g__phdd_kernel_bool
\bool_new:N \g__phdd_cs_break_bool
\bool_gset_true:N \g__phdd_cs_break_bool
```

\l__phdd_tmpa_tl

\l__phdd_tmpb_tl

\l__phdd_tmpa_int

\l__phdd_tmpa_seq

Some temporary variables.

```
\tl_new:N \l__phdd_tmpa_tl
\tl_new:N \l__phdd_tmpb_tl
\int_new:N \l__phdd_tmpa_int
\int_new:N \l__phdd_tmpa_seq
```

\l__phdd_names_block_tl

List of local sequence variables (produced through **__phdd_lseq_name:n**), one for each set of variants in a `function` or `macro` environment. More precisely these sequences are named after the base forms, such as **\clist_count:n** or **\clist_count:N** (which are not variants). Each of these sequences have the base name (without any signature) as their first item, followed by the list of variant's signatures, or **\scan_stop:** to denote the absence of signature (no colon).

```
\tl_new:N \l__phdd_names_block_tl
```

\g__phdd_variants_seq

Stores rather temporarily the list of variants (signatures only) of a function/-macro that is being documented. It is global because we need it to keep its value throughout cells of an alignment.

```
\seq_new:N \g__phdd_variants_seq
```

\l__phdd_names_verb_bool

Set to **true** if the main argument of a macro/function environment should be used as is, without removing any comma or space.

```
\bool_new:N \l__phdd_names_verb_bool
```

\l__phdd_names_seq

List of functions/environments/... appearing as arguments of a given `function` or `macro` environment. These are the names after conversion of `_@@` and `@@` to `__`⟨*module name*⟩ and other sanitizing.

```
\seq_new:N \l__phdd_names_seq
```

\g__phdd_nested_names_seq

Collects all macros in nested `macro` environments, to use them in the "End definition" text.

```
\seq_new:N \g__phdd_nested_names_seq
```

\l__phdd_index_macro_tl

\l__phdd_index_key_tl

\l__phdd_index_module_tl

\l__phdd_index_internal_bool

When analyzing a control sequence found within a `macrocode` environment, **\l__phdd_index_macro_tl** holds the control sequence (partially a string), **\l__phdd_index_key_tl** holds the future sort key in the index, and **\l__phdd_index_module_tl** is the subindex in which the control sequence should be listed. Finally, **\l__phdd_index_internal_bool** indicates when the control sequence is internal and should be indexed in a slightly different subindex.

```
\tl_new:N \l__phdd_index_macro_tl
\tl_new:N \l__phdd_index_key_tl
\tl_new:N \l__phdd_index_module_tl
\bool_new:N \l__phdd_index_internal_bool
```

\g__phdd_module_name_tl

The module name, set when reading a line `<@@=`⟨*module*⟩`>`.

```
\tl_new:N \g__phdd_module_name_tl
```

\c__phdd_iow_rule_tl

\c__phdd_iow_midrule_tl

40 equal signs.

```
\tl_const:Nn \c__phdd_iow_rule_tl
  { ======================================== }
\tl_const:Nn \c__phdd_iow_mid_rule_tl
  { ---------------------------------------- }
```

`\l__phdd_macro_box`

`\l__phdd_macro_index_box`

`\l__phdd_macro_int`

A vertical box in which the names given to the macro environment are typeset, a horizontal box in which we store the targets created by indexing commands, and the number of macros so far (including those from surrounding `macro` environments).

```
\box_new:N \l__phdd_macro_box
\box_new:N \l__phdd_macro_index_box
\int_new:N \l__phdd_macro_int
```

`\l__phdd_cmd_tl`

`\l__phdd_cmd_index_tl`

`\l__phdd_cmd_module_tl`

`\l__phdd_cmd_noindex_bool`

`\l__phdd_cmd_replace_bool`

Variables used to control the behaviour of **`\cmd`**, **`\cs`** and **`\tn`**.

```
\tl_new:N \l__phdd_cmd_tl
\tl_new:N \l__phdd_cmd_index_tl
\tl_new:N \l__phdd_cmd_module_tl
\bool_new:N \l__phdd_cmd_noindex_bool
\bool_new:N \l__phdd_cmd_replace_bool
```

`\l__phdd_in_implementation_bool`

This boolean is `true` within the `implementation` environment, and `false` anywhere else.

```
\bool_new:N \l__phdd_in_implementation_bool
```

`\g__phdd_typeset_documentation_bool`

`\g__phdd_typeset_implementation_bool`

These booleans control whether the documentation/implementation should be typeset. By default both should be.

```
\bool_new:N \g__phdd_typeset_documentation_bool
\bool_new:N \g__phdd_typeset_implementation_bool
\bool_set_true:N \g__phdd_typeset_documentation_bool
\bool_set_true:N \g__phdd_typeset_implementation_bool
```

\g__phdd_base_name_tl

\l__phdd_variants_prop

The name of the macro which is being documented (without its signature), and a property list mapping base forms of variants to all variants which have the same base form.

```
\tl_new:N \g__phdd_base_name_tl
\prop_new:N \l__phdd_variants_prop
```

\l__phdd_function_label_clist

\l__phdd_no_label_bool

Option of a `function` environment which replaces the label that would normally be inserted by labels for the given list of control sequences. This is only useful to avoid duplicate labels when a function's documentation appears multiple times.

```
\clist_new:N \l__phdd_function_label_clist
\bool_new:N \l__phdd_no_label_bool
```

\l__phdd_date_added_tl

\l__phdd_date_updated_tl

Values of some options of the `function` environment.

```
\tl_new:N \l__phdd_date_added_tl
\tl_new:N \l__phdd_date_updated_tl
```

\l__phdd_macro_argument_tl

Save the argument of a `macro` or `function` environment for use in error messages.

```
\tl_new:N \l__phdd_macro_argument_tl
% \int_new:N \c@CodelineNo
```

D.2.1 Variants and helpers

__phdd_tmpa:w
__phdd_tmpb:w

Auxiliary macros for temporary use.

```
\cs_new_eq:NN \__phdd_tmpa:w ?
\cs_new_eq:NN \__phdd_tmpb:w ?
```

(*End definition for* `\__phdd_tmpa:w` *and* `\__phdd_tmpb:w`.)

\seq_set_split:NoV
\str_case:fn
\tl_count:f
\tl_greplace_all:Nxn
\tl_greplace_all:Nno
ead_eq_charcode:oNTF
head_eq_charcode:oNT
head_eq_charcode:oNF
_head_eq_meaning:VNF
\tl_if_in:noT
\tl_if_in:ooTF
\tl_if_in:NoTF
\tl_if_in:NoT
\tl_if_in:NoF
\tl_remove_all:Nx
\tl_replace_all:Nxn
\tl_replace_all:Nnx
\tl_replace_all:Non

A few missing variants.

```
\cs_generate_variant:Nn \seq_set_split:Nnn { NoV }
\cs_generate_variant:Nn \seq_gput_right:Nn { Nf }
\cs_generate_variant:Nn \str_case:nn { fn }
\cs_generate_variant:Nn \tl_count:n { f }
\cs_generate_variant:Nn \tl_greplace_all:Nnn { Nx , Nno }
\cs_generate_variant:Nn \tl_if_empty:nTF { f }
\cs_generate_variant:Nn \tl_if_head_eq_charcode:nNTF { o }
```

```
\cs_generate_variant:Nn \tl_if_head_eq_charcode:nNT  { o }
\cs_generate_variant:Nn \tl_if_head_eq_charcode:nNF  { o }
\cs_generate_variant:Nn \tl_if_head_eq_meaning:nNF  { V }
\cs_generate_variant:Nn \tl_if_in:nnTF { no , oo }
\cs_generate_variant:Nn \tl_if_in:NnTF { No }
\cs_generate_variant:Nn \tl_if_in:NnT  { No }
\cs_generate_variant:Nn \tl_if_in:NnF  { No }
\cs_generate_variant:Nn \tl_remove_all:Nn   { Nx }
\cs_generate_variant:Nn \tl_replace_all:Nnn { Nx , Nnx, No , Nno }
\cs_generate_variant:Nn \tl_replace_once:Nnn { Noo }
\cs_generate_variant:Nn \tl_to_str:n { f , o }
\cs_generate_variant:Nn \prop_get:NnNTF { Nx }
\cs_generate_variant:Nn \prop_put:Nnn { Nx }
\cs_generate_variant:Nn \prop_gput:Nnn { NVx }
```

(*End definition for* `\seq_set_split:NoV` *and others.*)

`dd_if_almost_str:nTF`

Used to test if the argument of **\cmd** or other macros to be indexed is almost a string or not: for instance this is `false` if **#1** contains **\meta{...}**. The surprising **f**-expansion are there to cope with the case of **#1** starting with **\c_backslash_str** which should be expanded and considered to be "normal".

```
\prg_new_protected_conditional:Npnn \__phdd_if_almost_str:n #1 { TF , T , F }
  {
    \int_compare:nNnTF
      { \tl_count:n {#1} }
      < { \tl_count:f { \tl_to_str:f {#1} } }
      { \prg_return_false: }
      { \prg_return_true: }
  }
\cs_generate_variant:Nn \__phdd_if_almost_str:nT { V }
```

(*End definition for* `\__phdd_if_almost_str:nTF`.)

`__phdd_trim_right:Nn`
`__phdd_trim_right:No`

Removes all material after **#2** in the token list variable **#1**. Perhaps combine with **__phdd_key_trim_module:n**?

```
\cs_new_protected:Npn \__phdd_trim_right:Nn #1#2
  {
    \cs_set:Npn \__phdd_tmp:w ##1 #2 ##2 \q_stop { \exp_not:n {##1} }
    \tl_set:Nx #1 { \exp_after:wN \__phdd_tmp:w #1 #2 \q_stop }
  }
\cs_generate_variant:Nn \__phdd_trim_right:Nn { No }
```

(*End definition for* `\__phdd_trim_right:Nn`.)

`dd_str_if_begin:nnTF`
`dd_str_if_begin:ooTF`

True if the first string starts with the second.

```
\prg_new_protected_conditional:Npnn \__phdd_str_if_begin:nn #1#2 { TF , T , F }
  {
    \tl_if_in:ooTF
      { \exp_after:wN \scan_stop: \tl_to_str:n {#1} }
      { \exp_after:wN \scan_stop: \tl_to_str:n {#2} }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\prg_generate_conditional_variant:Nnn \__phdd_str_if_begin:nn
  { oo } { TF , T , F }
```

(*End definition for* `\__phdd_str_if_begin:nnTF`.)

phdd_replace_at_at:N
replace_at_at_aux:Nn

The goal is to replace `@@` by the current module name. We take advantage of this function to also detect internal macros. If there is no ⟨*module name*⟩, do nothing. Otherwise, sanitize the catcodes of `@` and `_`, temporarily change `@@@@` to `aa` with different catcodes and later to `@@`, and replace `__@@` and `_@@` and `@@` by `__`⟨*module name*⟩. The result contains `_` with category code letter because this is what the `macrocode` environment expects. Other use cases can apply `\tl_to_str:n` if needed. Note that we include spaces between the `@` in the code below, since it is also processed through the same replacement rules. Updated selection to include `!@@`. This must be used with l3doctripx otherwise the style file will not be able to do the conversion. This maybe can provide a generic package namespacing possibility.

U 2019-01-07

```
\cs_new_protected:Npn \__phdd_replace_at_at:N #1
  {
    \tl_if_empty:NF \g__phdd_module_name_tl
      {
        \exp_args:NNo \__phdd_replace_at_at_aux:Nn
          #1 \g__phdd_module_name_tl
      }
  }
\cs_new_protected:Npx \__phdd_replace_at_at_aux:Nn #1#2
  {
%    \tl_replace_all:Nnn #1 { \token_to_str:N @ } { @ }
%    \tl_replace_all:Nnn #1 { \token_to_str:N _ } { _ }
%    \tl_replace_all:Nnn #1 { \token_to_str:N ! } { ! }
%    \tl_replace_all:Nnn #1 { @ @ @ @ } { \token_to_str:N a a }
%    \tl_replace_all:Nnn #1 { _ _ @ @ } { _ _ #2 }
%    \tl_replace_all:Nnn #1 {   _ @ @ } { _ _ #2 }
%    \tl_replace_all:Nnn #1 {    !@ @ } {     #2 }
%    \tl_replace_all:Nnn #1 {     @ @ } { _ _ #2 }

   \tl_replace_all:Nnn #1 { \token_to_str:N @ } { @ }
    \tl_replace_all:Nnn #1 { \token_to_str:N _ } { _ }
    \tl_replace_all:Nnn #1 { \token_to_str:N ! } { ! }
    \tl_replace_all:Nnn #1 { @ @ @ @ } { \token_to_str:N a a }
    \tl_replace_all:Nnn #1 { _ _ @ @ } { _ _ #2 }
    \tl_replace_all:Nnn #1 {   _ @ @ } { _ _ #2 }
    \tl_replace_all:Nnn #1 {    !@ @ } {     #2 }
    \tl_replace_all:Nnn #1 {     @ @ } { _ _ #2 }

    \tl_replace_all:Nnn #1 { \token_to_str:N a a } { @ @ }
  }
```

(*End definition for* `\__phdd_replace_at_at:N` *and* `\__phdd_replace_at_at_aux:Nn`.)

d_detect_internals:N
tect_internals_aux:N
tect_internals_ok:NF

After splitting at each `__` and removing the leading item from the sequence (since it does not follow `__`), remove everything after any space or end-of-line to get a good approximation of the control sequence (for the warning message). Then check if that starts with something allowed: `@@` module name and `:` or `_`, or if the relevant boolean is set `kernel_` (it seems safe to assume we will not define a `\__kernel:...` command). For the message itself remove anything after any `_` or `:` (with either catcode) to get a guess of the module name.

```
\cs_new_protected:Npn \__phdd_detect_internals:N #1
  {
    \bool_if:NT \l__phdd_detect_internals_bool
      { \__phdd_detect_internals_aux:N #1 }
  }
\group_begin:
  \char_set_catcode_active:N \^^M
  \cs_new_protected:Npn \__phdd_detect_internals_aux:N #1
    {
      \tl_set_eq:NN \l__phdd_detect_internals_tl #1
      \tl_replace_all:Non \l__phdd_detect_internals_tl { \token_to_str:N _ } { _ }
      \seq_set_split:NnV \l__phdd_tmpa_seq { _ _ } \l__phdd_detect_internals_tl
      \seq_pop_left:NN \l__phdd_tmpa_seq \l__phdd_detect_internals_tl
      \seq_map_variable:NNn \l__phdd_tmpa_seq \l__phdd_detect_internals_tl
        {
          \__phdd_trim_right:No \l__phdd_detect_internals_tl
            \c_catcode_active_space_tl
          \__phdd_trim_right:Nn \l__phdd_detect_internals_tl ^^M
          \__phdd_if_detect_internals_ok:NF \l__phdd_detect_internals_tl
            {
              \tl_set_eq:NN \l__phdd_detect_internals_cs_tl \l__phdd_detect_internals_tl
              \__phdd_trim_right:Nn \l__phdd_detect_internals_tl _
              \__phdd_trim_right:Nn \l__phdd_detect_internals_tl :
              \__phdd_trim_right:No \l__phdd_detect_internals_tl { \token_to_str:N : }
              \msg_warning:nnxxx { phdd } { foreign-internal }
                { \tl_to_str:N \l__phdd_detect_internals_cs_tl }
                { \tl_to_str:N \l__phdd_detect_internals_tl }
                { \tl_to_str:N \g__phdd_module_name_tl }
            }
        }
    }
\group_end:
\prg_new_protected_conditional:Npnn \__phdd_if_detect_internals_ok:N #1 { F }
  {
    \__phdd_str_if_begin:ooTF {#1} { \g__phdd_module_name_tl _ }
      { \prg_return_true: }
      {
        \__phdd_str_if_begin:ooTF {#1} { \g__phdd_module_name_tl : }
          { \prg_return_true: }
          {
            \bool_if:NTF \g__phdd_kernel_bool
              {
                \__phdd_str_if_begin:ooTF {#1} { kernel _ }
                  { \prg_return_true: }
                  { \prg_return_false: }
              }
              { \prg_return_false: }
          }
      }
  }
```

(*End definition for* `\__phdd_detect_internals:N`, `\__phdd_detect_internals_aux:N`, *and* `\__phdd_if_detect_internals_ok:NF`.)

`ignature_base_form:n`
`ture_base_form_aux:n`
`ture_base_form_aux:`

Expands to the "base form" of the signature. For instance, given **noxcfvV** it

would obtain **nnnNnnn**, or given **ow** it would obtain **nw**. The loop stops at the first token that is not recognized; the rest is enclosed in **\exp_not:n**.

```
\cs_new:Npn \__phdd_signature_base_form:n #1
  { \__phdd_signature_base_form_aux:n #1 \q_stop }
\cs_new:Npn \__phdd_signature_base_form_aux:n #1
  {
    \str_case:nnTF {#1}
      {
        { N } { N }
        { c } { N }
        { n } { n }
        { o } { n }
        { f } { n }
        { x } { n }
        { V } { n }
        { v } { n }
      }
      { \__phdd_signature_base_form_aux:n }
      { \__phdd_signature_base_form_aux:w #1 }
  }
\cs_new:Npn \__phdd_signature_base_form_aux:w #1 \q_stop
  { \exp_not:n {#1} }
```

(*End definition for* `\__phdd_signature_base_form:n`, `\__phdd_signature_base_form_aux:n`, *and* `\__phdd_signature_base_form_aux:w`.)

`redicate_from_base:n`

Get predicate from a function's base name. The code is not broken by functions with no signature. The **n**-type version can be used for keys and other non-control sequences. The output after **x**-expansion is a string.

```
\cs_new:Npn \__phdd_predicate_from_base:n #1
  {
    \__phdd_get_function_name:n {#1}
    \tl_to_str:n { _p: }
    \__phdd_get_function_signature:n {#1}
  }
```

(*End definition for* `\__phdd_predicate_from_base:n`.)

`split_function_do:nn`
`split_function_do:on`
`_get_function_name:n`
`function_signature:n`
`plit_function_auxi:w`
`lit_function_auxii:w`

Similar to internal functions defined in **l3basics**, but here we operate on strings directly rather than control sequences.

```
\cs_new:Npn \__phdd_get_function_name:n #1
  { \__phdd_split_function_do:nn {#1} { \use_i:nnn } }
\cs_new:Npn \__phdd_get_function_signature:n #1
  { \__phdd_split_function_do:nn {#1} { \use_ii:nnn } }
\cs_set_protected:Npn \__phdd_tmpa:w #1
  {
    \cs_new:Npn \__phdd_split_function_do:nn ##1
      {
        \exp_after:wN \__phdd_split_function_auxi:w
        \tl_to_str:n {##1} \q_mark \c_true_bool
        #1 \q_mark \c_false_bool
        \q_stop
      }
    \cs_new:Npn \__phdd_split_function_auxi:w
```

```
      ##1 #1 ##2 \q_mark ##3##4 \q_stop ##5
      { \__phdd_split_function_auxii:w {##5} ##1 \q_mark \q_stop {##2} ##3 }
    \cs_new:Npn \__phdd_split_function_auxii:w
      ##1##2 \q_mark ##3 \q_stop
      { ##1 {##2} }
  }
\exp_args:No \__phdd_tmpa:w { \token_to_str:N : }
\cs_generate_variant:Nn \__phdd_split_function_do:nn { o }
```

(*End definition for* `\__phdd_split_function_do:nn` *and others.*)

`phdd_key_get_base:nN`

Get the base form of a function and store it. As part of getting the base form, change trailing **T** or **F** to **TF**, skipping that change if the function contains no colon to avoid changing for instance some names ending in PDF or similar. The various letters **z** serve as end-delimiters different from any outcome of **\tl_to_str:n**.

```
\cs_new_protected:Npn \__phdd_key_get_base:nN #1#2
  {
    \__phdd_if_almost_str:nTF {#1}
      {
        \__phdd_key_get_base_TF:nN {#1} \l__phdd_tmpa_tl
        \tl_set:Nx #2
          { \__phdd_split_function_do:on \l__phdd_tmpa_tl { \__phdd_base_form_aux:nnN }
      }
      { \tl_set:Nn #2 {#1} }
  }
\cs_new:Npx \__phdd_key_get_base_TF:nN #1#2
  {
    \tl_set:Nx #2 { \exp_not:N \tl_to_str:n {#1} }
    \tl_if_in:NoF #2 { \tl_to_str:n {:} }
      { \exp_not:N \prg_break: }
    \tl_if_in:onT { #2 z } { \tl_to_str:n {TF} z }
      { \exp_not:N \prg_break: }
    \tl_if_in:onT { #2 z } { \tl_to_str:n {T} z }
      {
        \tl_put_right:Nn #2 { \tl_to_str:n {F} }
        \exp_not:N \prg_break:
      }
    \tl_if_in:onT { #2 z } { \tl_to_str:n {F} z }
      {
        \tl_put_right:Nn #2 { z }
        \tl_replace_once:Nnn #2 { \tl_to_str:n {F} z } { \tl_to_str:n {TF} }
        \exp_not:N \prg_break:
      }
    \exp_not:N \prg_break_point:
  }
\cs_new:Npn \__phdd_base_form_aux:nnN #1#2#3
  {
    \exp_not:n {#1}
    \bool_if:NT #3
      {
        \token_to_str:N :
        \str_if_eq:nnTF { #1 ~ } { \exp_args }
          { \exp_not:n {#2} }
```

```
            { \__phdd_signature_base_form:n {#2} }
        }
    }
```

(*End definition for* `\__phdd_key_get_base:nN`.)

`orm_signature_do:nnn`

Do `#2{#1}` if there is no signature, or if `#1` contains two colons in a row (this covers the weird function `\::N` and so on). Otherwise apply `#3` with the following two arguments: the base form of `#1`, and the original signature with an extra pair of braces.

```
\cs_new_protected:Npn \__phdd_base_form_signature_do:nnn #1#2#3
  {
    \__phdd_split_function_do:nn {#1}
      { \__phdd_base_form_aux:nnnnnN {#1} {#2} {#3} }
  }
\cs_new_protected:Npn \__phdd_base_form_aux:nnnnnN #1#2#3#4#5#6
  {
    \bool_if:NTF #6
      {
        \tl_if_head_eq_charcode:nNTF {#4} :
          { #2 {#1} }
          {
            \use:x
              {
                \exp_not:n {#3}
                { \__phdd_base_form_aux:nnN {#4} {#5} #6 }
              }
                {#4} {#5}
          }
      }
      { #2 {#1} }
  }
```

(*End definition for* `\__phdd_base_form_signature_do:nnn`.)

`d_date_compare_p:nNn`
`d_date_compare:nNnTF`
`_compare_aux:nnnNnnn`
`d_date_compare_aux:w`

Expects `#1` and `#3` to be dates in the format YYYY-MM-DD (but accepts YYYY or YYYY-MM too). Compares them using `#2` (one of `<`, `=`, `>`), filling in zeros for missing data.

```
\prg_new_conditional:Npnn \__phdd_date_compare:nNn #1#2#3 { TF , T , F , p }
  { \__phdd_date_compare_aux:w #1--- \q_mark #2 #3--- \q_stop }
\cs_new:Npn \__phdd_date_compare_aux:w
    #1 - #2 - #3 - #4 \q_mark #5 #6 - #7 - #8 - #9 \q_stop
  {
    \__phdd_date_compare_aux:nnnNnnn
      { \tl_if_empty:nTF {#1} { 0 } {#1} }
      { \tl_if_empty:nTF {#2} { 0 } {#2} }
      { \tl_if_empty:nTF {#3} { 0 } {#3} }
      #5
      { \tl_if_empty:nTF {#6} { 0 } {#6} }
      { \tl_if_empty:nTF {#7} { 0 } {#7} }
      { \tl_if_empty:nTF {#8} { 0 } {#8} }
  }
\cs_new:Npn \__phdd_date_compare_aux:nnnNnnn #1#2#3#4#5#6#7
  {
```

```
    \int_compare:nNnTF {#1} = {#5}
      {
        \int_compare:nNnTF {#2} = {#6}
          {
            \int_compare:nNnTF {#3} #4 {#7}
              { \prg_return_true: } { \prg_return_false: }
          }
          {
            \int_compare:nNnTF {#2} #4 {#6}
              { \prg_return_true: } { \prg_return_false: }
          }
      }
      {
        \int_compare:nNnTF {#1} #4 {#5}
          { \prg_return_true: } { \prg_return_false: }
      }
    \use_none:n
    \q_stop
  }
```

(*End definition for* `\__phdd_date_compare:nNnTF`, `\__phdd_date_compare_aux:nnnNnnn`, *and* `\__phdd_date_compare_aux:w`.)

`\__phdd_gprop_name:n`
`\__phdd_lseq_name:n`

We need to keep track of some information about control sequences (and other strings) that are being (or have been) documented. Some is stored into global props and some into local seqs, whose name does not follow conventions: it is **\g__phdd** or **\l__phdd** followed by a space and by the string, which can be arbitrary. We cannot reasonably use a single big **prop** for speed reasons.

```
\cs_new:Npn \__phdd_gprop_name:n #1 { g__phdd ~ \tl_to_str:n {#1} }
\cs_new:Npn \__phdd_lseq_name:n #1 { l__phdd ~ \tl_to_str:n {#1} }
```

(*End definition for* `\__phdd_gprop_name:n` *and* `\__phdd_lseq_name:n`.)

D.2.2 Messages

```
\msg_new:nnnn { phdd } { no-signature-TF }
  { Function/macro~'#1'~cannot~be~turned~into~a~conditional. }
  {
    A~function~or~macro~environment~with~option~pTF,~TF~or~noTF~
    received~the~argument~'#1'.~This~function's~name~has~no~
    ':'~hence~it~is~not~clear~where~to~add~'_p'~or~'TF'.~
    Please~follow~expl3~naming~conventions.
  }
\msg_new:nnn { phdd } { deprecated-function }
  { The~deprecated~function(s)~'#1'~should~have~been~removed~on~#2. }

\msg_new:nnn { phdd } { date-format }
  { The~date~'#1'~should~be~given~in~YYYY-MM-DD~format. }

\msg_new:nnn { phdd } { syntax-nested-function }
  {
    The~'syntax'~environment~should~be~used~in~the~
    innermost~'function'~environment.
  }
```

```
\msg_new:nnn { phdd } { multiple-syntax }
  {
    The~'syntax'~environment~should~only~be~used~once~in~
    a~'function'~environment.
  }
\msg_new:nnn { phdd } { deprecated-option }
  { The~option~'#1'~has~been~deprecated~for~'#2'. }

\msg_new:nnn { phdd } { foreign-internal }
  {
    A~control~sequence~of~the~form~'..._#1'~was~used.~
    It~should~only~be~used~in~the~module~'#2'
    \tl_if_empty:nF {#3} { ,~not~in~'#3' } .
  }
```

D.2.3 Configuration and tweaks

`\MakePrivateLetters`

A few more letters are "private" in a LaTeX3 programming environment.

```
\cs_gset:Npn \MakePrivateLetters
  {
    \char_set_catcode_letter:N \@
    \char_set_catcode_letter:N \_
    \char_set_catcode_letter:N \:
  }
```

(*End definition for* `\MakePrivateLetters`.)

`CodelineNo`

Some configurations which have to do with line numbering. This is now provisioned through a key value interface.

```
\setcounter{StandardModuleDepth}{1}
%\@addtoreset{CodelineNo}{part}
%\tl_replace_once:Nnn \theCodelineNo
%  { \HDorg@theCodelineNo }
%  { \textcolor[gray]{0.5} { \sffamily\tiny\arabic{CodelineNo} } }
```

(*End definition for* `CodelineNo`.)

`\verbatim` `\endverbatim`

In `.dtx` documents, the `verbatim` environment adds extra space because it only removes the first "%" sign, and not the indentation (typically a space). Fix it with **fancyvrb**:

```

% \cs_gset_eq:NN \verbatim \Verbatim
% \cs_gset_eq:NN \endverbatim \endVerbatim
```

(*End definition for* `\verbatim` *and* `\endverbatim`.)

`\ifnot@excluded`

This function tests whether a macro name stored in `\macro@namepart` was excluded from indexing by `\DoNotIndex`. Rather than trying to fix catcodes that come into here, turn everything to string catcodes. This is somewhat inefficient as we could have ensured that `\index@excludelist` has string catcodes in the first place.

```
\cs_set_protected:Npn \ifnot@excluded
  {
    \exp_args:Nxx \expanded@notin
```

```
      { \c_backslash_str \tl_to_str:N \macro@namepart , }
      { \exp_args:NV \tl_to_str:n \index@excludelist }
  }
```

(*End definition for* `\ifnot@excluded`.)

`\pdfstringnewline`
`\__pdfstring_newline:w`

We avoid some hyperref warnings by making `\\` (almost) trivial in bookmarks: more precisely it might be used with a star and an optional argument, which we thus remove using an **xparse** expandable command. Since there cannot be trailing optional arguments, pick up an extra mandatory one and put it back.

```
\cs_new:Npn \pdfstringnewline { : ~ }
\DeclareExpandableDocumentCommand
  { \__phdd_pdfstring_newline:w } { s o m } { \pdfstringnewline #3 }
\pdfstringdefDisableCommands
  { \cs_set_eq:NN \\ \__phdd_pdfstring_newline:w }
```

(*End definition for* `\pdfstringnewline` *and* `\__phdd_pdfstring_newline:w`.)

`\cmd`
`\cs`
`\tn`

They rely on a common auxiliary `\__phdd_cmd:nn` which receives as arguments the options and some tokens whose string representation starts with a backslash (to support cases such as `\cs{pkg_\ldots}`, we do not turn the whole argument into a string).

```
\DeclareDocumentCommand \cmd { O{} m }
  { \__phdd_cmd:no {#1} { \token_to_str:N #2 } }

\DeclareDocumentCommand \cs  { O{} m }
  { \__phdd_cmd:no {#1} { \c_backslash_str #2 } }

\DeclareDocumentCommand \tn  { O{} m }
  {
    \__phdd_cmd:no
      { module = TeX , replace = false , #1 }
      { \c_backslash_str #2 }
  }
```

(*End definition for* `\cmd`, `\cs`, *and* `\tn`.)

`\meta`

A document-level command.

```
\DeclareDocumentCommand \meta { m }
  { \__phdd_meta:n {#1} }
```

(*End definition for* `\meta`.)

`\__phdd_pdfstring_cmd:w`
`\__phdd_pdfstring_cs:w`
`\__phdd_pdfstring_meta:w`

To work within a bookmark, these commands must be expandable.

```
\DeclareExpandableDocumentCommand
  { \__phdd_pdfstring_cmd:w } { o m } { \token_to_str:N #2 }
\DeclareExpandableDocumentCommand
  { \__phdd_pdfstring_cs:w }  { o m } { \textbackslash \tl_to_str:n {#2} }
\cs_new:Npn \__phdd_pdfstring_meta:w #1
  { < \tl_to_str:n {#1} > }
\pdfstringdefDisableCommands
  {
    \cs_set_eq:NN \cmd  \__phdd_pdfstring_cmd:w
    \cs_set_eq:NN \cs   \__phdd_pdfstring_cs:w
    \cs_set_eq:NN \tn   \__phdd_pdfstring_cs:w
```

```
    \cs_set_eq:NN \meta \__phdd_pdfstring_meta:w
  }
```

(*End definition for* `\__phdd_pdfstring_cmd:w`, `\__phdd_pdfstring_cs:w`, *and* `\__phdd_pdfstring_meta:w`.)

`\Arg` `\marg` `\oarg` `\parg`

`\marg{text}` prints {⟨*text*⟩}, "mandatory argument". `\oarg{text}` prints [⟨*text*⟩], "optional argument". `\parg{te,xt}` prints (⟨*te,xt*⟩), "picture mode argument". Finally, `\Arg` is the same as `\marg`.

```
\newcommand\Arg[1]
  { \texttt{\char`\{} \meta{#1} \texttt{\char`\}} }
\providecommand\marg[1]{ \Arg{#1} }
\providecommand\oarg[1]{ \texttt[ \meta{#1} \texttt] }
\providecommand\parg[1]{ \texttt( \meta{#1} \texttt) }
```

(*End definition for* `\Arg` *and others.*)

`\file` `\env` `\pkg` `\cls`

This list may change...this is just my preference for markup.

```
\DeclareRobustCommand \file {\nolinkurl}
\DeclareRobustCommand \env {\texttt}

%\DeclareRobustCommand \pkg {\textsf}
%\DeclareRobustCommand \cls {\textsf}

```

(*End definition for* `\file` *and others.*)

`\EnableDocumentation` `EnableImplementation` `DisableDocumentation` `isableImplementation`

Control whether to typeset the documentation/implementation or not. These simply set two switches.

```
\NewDocumentCommand \EnableDocumentation { }
  { \bool_gset_true:N \g__phdd_typeset_documentation_bool }
\NewDocumentCommand \EnableImplementation { }
  { \bool_gset_true:N \g__phdd_typeset_implementation_bool }
\NewDocumentCommand \DisableDocumentation { }
  { \bool_gset_false:N \g__phdd_typeset_documentation_bool }
\NewDocumentCommand \DisableImplementation { }
  { \bool_gset_false:N \g__phdd_typeset_implementation_bool }
```

(*End definition for* `\EnableDocumentation` *and others.*)

`documentation` `implementation`

If the documentation/implementation should be typeset, then simply set the boolean `\l__phdd_in_implementation_bool` which indicates whether we are within the implementation section. Otherwise use `\comment` (and a paired `\endcomment`).

```
\NewDocumentEnvironment { documentation } { }
  {
    \bool_if:NTF \g__phdd_typeset_documentation_bool
      { \bool_set_false:N \l__phdd_in_implementation_bool }
      { \comment }
  }
  { \bool_if:NF \g__phdd_typeset_documentation_bool { \endcomment } }
\NewDocumentEnvironment { implementation } { }
  {
    \bool_if:NTF \g__phdd_typeset_implementation_bool
```

```
      { \bool_set_true:N \l__phdd_in_implementation_bool }
      { \comment }
  }
  { \bool_if:NF \g__phdd_typeset_implementation_bool { \endcomment } }
```

variable The `variable` environment behaves as a `function` or `macro` environment depending on the part of the document.

```
\DeclareDocumentEnvironment { variable } { O{} +v }
  {
    \bool_if:NTF \l__phdd_in_implementation_bool
      { \__phdd_macro:nnw { var , #1 } {#2} }
      { \__phdd_function:nnw {#1} {#2} }
  }
  {
    \bool_if:NTF \l__phdd_in_implementation_bool
      { \__phdd_macro_end: }
      { \__phdd_function_end: }
  }
```

function macro Environment for documenting function(s), and environment for documenting the implementation of a macro.

```
\DeclareDocumentEnvironment { function } { O{} +v }
  { \__phdd_function:nnw {#1} {#2} }
  { \__phdd_function_end: }
\DeclareDocumentEnvironment { macro } { O{} +v }
  { \__phdd_macro:nnw {#1} {#2} }
  { \__phdd_macro_end: }
```

syntax Syntax block placed next to the list of functions to illustrate their use. TODO: test that the `syntax` environment is only used inside the `function` environment, and that it only appears once.

```
\NewDocumentEnvironment { syntax } { }
  { \__phdd_syntax:w }
  {
    \__phdd_syntax_end:
    \ignorespacesafterend
  }
```

texnote Used to describe information destined to TEX experts only.

```
\NewDocumentEnvironment { texnote } { }
  {
    \endgraf
    \vspace{3mm}
    \small\textbf{\TeX~hackers~note:}
  }
  {
    \vspace{3mm}
  }
```

\CodedocExplain \CodedocExplainEXP \CodedocExplainREXP \CodedocExplainTF Explanation of stars and **TF** notations, for use in third-party packages.

```
\NewDocumentCommand { \CodedocExplain } { }
  { \CodedocExplainEXP \ \CodedocExplainREXP \ \CodedocExplainTF }
\NewDocumentCommand { \CodedocExplainEXP } { }
```

```
  {
    \raisebox{\baselineskip}[0pt][0pt]{\hypertarget{expstar}{}}%
    \__phdd_typeset_exp:\ indicates~fully~expandable~functions,~which~
    can~be~used~within~an~\texttt{x}-type~argument~(in~plain~
    \TeX{}~terms,~inside~an~\cs{edef}),~as~well~as~within~an~
    \texttt{f}-type~argument.
  }
\NewDocumentCommand { \CodedocExplainREXP } { }
  {
    \raisebox{\baselineskip}[0pt][0pt]{\hypertarget{rexpstar}{}}%
    \__phdd_typeset_rexp:\ indicates~
    restricted~expandable~functions,~which~can~be~used~within~an~
    \texttt{x}-type~argument~but~cannot~be~fully~expanded~within~an~
    \texttt{f}-type~argument.
  }
\NewDocumentCommand { \CodedocExplainTF } { }
  {
    \raisebox{\baselineskip}[0pt][0pt]{\hypertarget{explTF}{}}%
    \__phdd_typeset_TF:\ indicates~conditional~(\texttt{if})~functions~
    whose~variants~with~\texttt{T},~\texttt{F}~and~\texttt{TF}~
    argument~specifiers~expect~different~
    \enquote{true}/\enquote{false}~branches.
  }
```

(*End definition for* `\CodedocExplain` *and others.*)

D.2.4 Implementing text markup

Keys for **\cmd**, **\cs** and **\tn**.

```
\keys_define:nn { phdd/cmd }
  {
    index     .tl_set:N     = \l__phdd_cmd_index_tl        ,
    module    .tl_set:N     = \l__phdd_cmd_module_tl       ,
    no-index  .bool_set:N   = \l__phdd_cmd_noindex_bool    ,
    replace   .bool_set:N   = \l__phdd_cmd_replace_bool    ,
  }
```

`\__phdd_cmd:nn`
`\__phdd_cmd:no`

Apply the key–value ⟨*options*⟩ **#1** after setting some default values. Then (unless **replace=false**) replace **@@** in **#2**, which is a bit tricky: the **_** must be given the catcode expected by **__phdd_replace_at_at:N**, but should be reverted to their original catcode (normally active, needed for line-breaking) without rescanning the whole argument. Then typeset the command in **\verbatim@font**, after turning it to harmless characters if needed (and keeping the underscore breakable); in any case, spaces must be turned into **\@xobeysp** and we must use **\@** to avoid longer spaces after a control sequence that ends for instance with a colon (empty signature). Finally, produce an index entry. Indexing is suppressed when **\l__phdd_cmd_noindex_bool** is true.

```
\cs_new_protected:Npn \__phdd_cmd:nn #1#2
  {
    \bool_set_false:N \l__phdd_cmd_noindex_bool
    \bool_set_true:N \l__phdd_cmd_replace_bool
    \tl_set:Nn \l__phdd_cmd_index_tl { \q_no_value }
```

```
    \tl_set:Nn \l__phdd_cmd_module_tl { \q_no_value }
    \keys_set:nn { phdd/cmd } {#1}
    \tl_set:Nn \l__phdd_cmd_tl {#2}
    \bool_if:NT \l__phdd_cmd_replace_bool
      {
        \tl_set_rescan:Nnn \l__phdd_tmpb_tl { } { _ }
        \tl_replace_all:Non \l__phdd_cmd_tl \l__phdd_tmpb_tl { _ }
        \__phdd_replace_at_at:N \l__phdd_cmd_tl
        \tl_replace_all:Nno \l__phdd_cmd_tl { _ } \l__phdd_tmpb_tl
      }
```

Typesetting Note the replacement for the underscore is to permit linebreaks. The `underscore` package adds the linebreak, and the regex results in applying the breakable underscore only to the *last* of a run of underscores, and not if the underscore follows a backslash.

```
\def\BreakableUnderscore{\leavevmode\nobreak\hskip\z@skip
 %\ifx\f@family\ttdefault \string \else
 \textunderscore
 %\fi
 \@dischyph\nobreak\hskip\z@skip}
    \mode_if_math:T { \mbox }
      {
        \verbatim@font
        \__phdd_if_almost_str:VT \l__phdd_cmd_tl
          {
            \tl_set:Nx \l__phdd_cmd_tl { \tl_to_str:N \l__phdd_cmd_tl }
            \bool_if:NT \g__phdd_cs_break_bool
              {
                % underscore is replaced with breakable underscore.
                \regex_replace_all:nnN
                  {([^\\])_([^\_])}
                  {\1\c{BreakableUnderscore}\2}
                  \l__phdd_cmd_tl
              }
          }
        \tl_replace_all:Nnn \l__phdd_cmd_tl { ~ } { \@xobeysp }
        \l__phdd_cmd_tl
        \@
      }
```

Indexing

```
    \bool_if:NF \l__phdd_cmd_noindex_bool
      {
        \quark_if_no_value:NF \l__phdd_cmd_index_tl
          {
            \tl_set:Nx \l__phdd_cmd_tl
              { \c_backslash_str \exp_not:o { \l__phdd_cmd_index_tl } }
          }

        \exp_args:No \__phdd_key_get:n { \l__phdd_cmd_tl }
        \quark_if_no_value:NF \l__phdd_cmd_module_tl
          {
```

```
            \tl_set:Nx \l__phdd_index_module_tl
              { \tl_to_str:N \l__phdd_cmd_module_tl }
          }
        \__phdd_special_index_module:ooonN
          { \l__phdd_index_key_tl }
          { \l__phdd_index_macro_tl }
          { \l__phdd_index_module_tl }
          { usage }
          \l__phdd_index_internal_bool
      }
  }
\cs_generate_variant:Nn \__phdd_cmd:nn { no }
```

(*End definition for* `\__phdd_cmd:nn`.)

`\__phdd_meta:n`
`phdd_ensuremath_sb:n`
`phdd_meta_original:n`

Store `#1` in `\l__phdd_tmpa_tl` and replaces every underscore, regardless of its category ("math toggle", "alignment", "superscript", "subscript", "letter", "other", or "active") by `\__phdd_ensuremath_sb:n` (which creates math subscripts), then runs the code used for `\meta` in **doc.sty**.

```
\cs_new_protected:Npn \__phdd_meta:n #1
  {
    \tl_set:Nn \l__phdd_tmpa_tl {#1}
    \tl_map_inline:nn
      { { 3 } { 4 } { 7 } { 8 } { 11 } { 12 } { 13 } }
      {
        \tl_set_rescan:Nnn \l__phdd_tmpb_tl
          { \char_set_catcode:nn { `_ } {##1} } { _ }
        \tl_replace_all:Non \l__phdd_tmpa_tl \l__phdd_tmpb_tl
          { \__phdd_ensuremath_sb:n }
      }
    \exp_args:NV \__phdd_meta_original:n \l__phdd_tmpa_tl
  }

\cs_new_protected:Npn \__phdd_ensuremath_sb:n #1
  { \ensuremath { \sb {#1} } }

% Orginal meta
\cs_new_protected:Npn \__phdd_meta_original:n #1
  {
    \ensuremath \langle
    \mode_if_math:T { \nfss@text }
    {
      \meta@font@select
      \edef \meta@hyphen@restore
        { \hyphenchar \the \font \the \hyphenchar \font }
      \hyphenchar \font \m@ne
      \language \l@nohyphenation
      \color{themeta}#1 \/
      \meta@hyphen@restore
    }
    \ensuremath \rangle
  }
```

(*End definition for* `\__phdd_meta:n`, `\__phdd_ensuremath_sb:n`, *and* `\__phdd_meta_original:n`.)

D.2.40 Common between `macro` and `function`

__phdd_typeset_exp:
__phdd_typeset_rexp:
__phdd_typeset_TF:
__phdd_typeset_aux:n

Used by **__phdd_macro_single:nNN** and in the `function` environment to typeset conditionals and auxiliary functions.

```
%
\cs_new_protected:Npn \__phdd_typeset_exp: % check
  { \hyperlink{expstar} {$\star$} }

\cs_new_protected:Npn \__phdd_typeset_rexp:
  { \hyperlink{rexpstar} {\ding{73}} } % hollow star

\cs_new_protected:Npn \__phdd_typeset_TF:
  {
    \hyperlink{explTF}
      {
        \color{themacrocolor} % was black
        \itshape TF
        \makebox[0pt][r]
          {
            \color{red}
            \underline { \phantom{\itshape TF} \kern-0.1em }
          }
      }
  }
\cs_new_protected:Npn \__phdd_typeset_aux:n #1
  {
    {
    %\color[gray]{0.5}
    %YL
    \color{themacrocolor} #1
    }
  }
```

(*End definition for* `\__phdd_typeset_exp:` *and others.*)

_get_hyper_target:nN
_get_hyper_target:oN
_get_hyper_target:xN

Create a **hyperref** anchor from a macro name **#1** and stores it in the token list variable **#2**. For instance, **\prg_replicate:nn** gives **doc/function//prg/replicate:nn**.

```
\cs_new_protected:Npn \__phdd_get_hyper_target:nN #1#2
  {
    \tl_set:Nx #2 { \tl_to_str:n {#1} }
    \tl_replace_all:Nxn #2 { \c_underscore_str } { / }
    \tl_remove_all:Nx   #2 { \c_backslash_str }
    \tl_put_left:Nn #2 { doc/function// }
  }
\cs_generate_variant:Nn \__phdd_get_hyper_target:nN { o , x }
```

(*End definition for* `\__phdd_get_hyper_target:nN`.)

hdd_names_get_seq:nN

The argument **#1** (argument of a **function** or **macro** environment) has catcodes 10 (space), 12 (other) and 13 (active). Sanitize catcodes. If the **verb** option was used, output a one-item sequence. Otherwise, remove any "%" character at the beginning of a line. Remove tabs and newlines. Finally, convert **_@@**

and @@ to __⟨*module name*⟩ (if it is non-empty). At this point, **\l__phdd_tmpa_tl** contains a comma-delimited list of names, where @ and _ have category code letter. Turn it to a string, parse it as a comma-delimited list (in particular this removes spaces), and output a sequence of function/macro names.

```
\cs_new_protected:Npn \__phdd_names_get_seq:nN #1#2
  {
    \tl_set:Nx \l__phdd_tmpa_tl { \tl_to_str:n {#1} }
    \bool_if:NTF \l__phdd_names_verb_bool
      {
        \seq_clear:N #2
        \seq_put_right:NV #2 \l__phdd_tmpa_tl
      }
      {
        \tl_remove_all:Nx \l__phdd_tmpa_tl
          { \iow_char:N \^^M \c_percent_str }
        \tl_remove_all:Nx \l__phdd_tmpa_tl { \tl_to_str:n { ^ ^ A } }
        \tl_remove_all:Nx \l__phdd_tmpa_tl { \iow_char:N \^^I }
        \tl_remove_all:Nx \l__phdd_tmpa_tl { \iow_char:N \^^M }
        \__phdd_detect_internals:N \l__phdd_tmpa_tl
        \__phdd_replace_at_at:N \l__phdd_tmpa_tl
        \exp_args:NNx \seq_set_from_clist:Nn #2
          { \tl_to_str:N \l__phdd_tmpa_tl }
      }
  }
```

(*End definition for* `\__phdd_names_get_seq:nN`.)

__phdd_names_parse:
__phdd_names_parse_one:n

The goal is to group variants together. We populate **\l__phdd_names_block_tl** with local sequence variable named with **__phdd_lseq_name:n** after the base forms. When encountering a new base form, set the corresponding local sequence to hold the ⟨*base name*⟩ (stripped of the signature) and add the local sequence to the list **\l__phdd_names_block_tl**. In all cases append the signature to the local sequence, which thus takes the form ⟨*base name*⟩, ⟨*signature*$_1$⟩, ⟨*signature*$_2$⟩ and so on. If the original function had no signature (no colon) then use **\scan_stop:** as the signature (there can be no variant). We special case commands **#1** starting with **\::**, namely weird functions named **\::N** and the like.

```
\cs_new_protected:Npn \__phdd_names_parse:
  {
    \tl_clear:N \l__phdd_names_block_tl
    \seq_map_function:NN
      \l__phdd_names_seq
      \__phdd_names_parse_one:n
  }
\cs_new_protected:Npn \__phdd_names_parse_one:n #1
  {
    \__phdd_split_function_do:nn {#1}
      { \__phdd_names_parse_one_aux:nnNn }
    {#1}
  }
\cs_new_protected:Npn \__phdd_names_parse_one_aux:nnNn #1#2#3#4
  {
    \bool_if:NTF #3
```

```
      {
        \tl_if_head_eq_charcode:nNTF {#2} :
          { \__phdd_names_parse_aux:nnn {#4} {#4} { \scan_stop: } }
          {
            \exp_args:Nx \__phdd_names_parse_aux:nnn
              { \__phdd_base_form_aux:nnN {#1} {#2} #3 }
              {#1} {#2}
          }
      }
      {
        \bool_if:NT \l__phdd_macro_TF_bool
          { \msg_error:nnx { l3doc } { no-signature-TF } {#4} }
        \__phdd_names_parse_aux:nnn {#4} {#4} { \scan_stop: }
      }
  }
\cs_new_protected:Npn \__phdd_names_parse_aux:nnn #1
  { \exp_args:Nc \__phdd_names_parse_aux:Nnn { \__phdd_lseq_name:n {#1} } }
\cs_new_protected:Npn \__phdd_names_parse_aux:Nnn #1#2#3
  {
    \tl_if_in:NnF \l__phdd_names_block_tl {#1}
      {
        \tl_put_right:Nn \l__phdd_names_block_tl {#1}
        \seq_clear_new:N #1
        \seq_put_right:Nn #1 {#2}
      }
    \seq_put_right:Nn #1 {#3}
  }
```

(*End definition for* `\__phdd_names_parse:` *and* `\__phdd_names_parse_one:n`.)

`\__phdd_names_typeset:` `\__phdd_names_typeset_auxi:n`

This code is in particular used when typesetting function names in a `function` environment. The mapping to **`\l__phdd_names_block_tl`** cannot use **`\tl_map_inline:Nn`** because the code following `\\` would not be expandable, thus breaking **`\bottomrule`**.

Call **`\__phdd_names_typeset_auxi:n`** on each local sequence (which holds a set of variants). The first step is to pop the base form and change spaces to category other so that they get displayed eventually. Then store the variants in **`\g__phdd_variants_seq`**, remove the first, which will be displayed more prominently, and reconstruct the actual name, passing it to **`\__phdd_names_typeset_auxii:n`**.

```
\cs_new_protected:Npn \__phdd_names_typeset:
  {
    \tl_map_function:NN \l__phdd_names_block_tl
      \__phdd_names_typeset_auxi:n
  }
\cs_new_protected:Npn \__phdd_names_typeset_auxi:n #1
  {
    \seq_pop:NN #1 \l__phdd_tmpa_tl
    \tl_gset_eq:NN \g__phdd_base_name_tl \l__phdd_tmpa_tl
    \tl_greplace_all:Nno \g__phdd_base_name_tl
      { ~ } { \c_catcode_other_space_tl }
    \seq_get:NN #1 \l__phdd_tmpa_tl
    \str_if_eq:VnTF \l__phdd_tmpa_tl { \scan_stop: }
      {
```

```
        \seq_gclear:N \g__phdd_variants_seq
        \__phdd_names_typeset_auxii:x { \g__phdd_base_name_tl }
      }
      {
        \seq_gset_eq:NN \g__phdd_variants_seq #1
        \seq_gpop:NN \g__phdd_variants_seq \l__phdd_tmpb_tl
        \__phdd_names_typeset_auxii:x
          { \g__phdd_base_name_tl : \l__phdd_tmpb_tl }
      }
  }
```

(*End definition for* `\__phdd_names_typeset:` *and* `\__phdd_names_typeset_auxi:n`.)

`ames_typeset_auxii:n`
`ames_typeset_auxii:x`

In case the option **pTF** was given, typeset predicates before the **TF** functions. In case the option **noTF** was given, typeset the non-**TF** function as well. Pass the relevant boolean in both cases to control whether to append **TF**.

```
\cs_new_protected:Npn \__phdd_names_typeset_auxii:n #1
  {
    \bool_if:NT \l__phdd_macro_pTF_bool
      {
        \__phdd_names_typeset_block:xN
          { \__phdd_predicate_from_base:n {#1} }
          \c_false_bool
      }
    \bool_if:NT \l__phdd_macro_noTF_bool
      { \__phdd_names_typeset_block:nN {#1} \c_false_bool }
    \__phdd_names_typeset_block:nN {#1} \l__phdd_macro_TF_bool
  }
\cs_generate_variant:Nn \__phdd_names_typeset_auxii:n { x }
```

(*End definition for* `\__phdd_names_typeset_auxii:n`.)

`mes_typeset_block:nN`
`mes_typeset_block:xN`

Names in `function` and `macro` environments are typeset differently. To distinguish the two note that **\l__phdd_nested_macro_int** is at least one when in an `macro` environment (we assume `function` is not nested inside it). A block is a function with all its variants.

```
\cs_new_protected:Npn \__phdd_names_typeset_block:nN
  {
    \int_compare:nNnTF \l__phdd_nested_macro_int = 0
      { \__phdd_typeset_function_block:nN }
      { \__phdd_macro_typeset_block:nN }
  }
\cs_generate_variant:Nn \__phdd_names_typeset_block:nN { x }
```

(*End definition for* `\__phdd_names_typeset_block:nN`.)

`f_macro_internal_p:n`
`f_macro_internal:nTF`
`macro_internal_aux:w`

Determines whether the given macro should be considered internal or public. If an option such as **int** was given then the answer is **\l__phdd_macro_internal_bool**, otherwise check for whether the macro name contains **__**.

```
\prg_new_conditional:Npnn \__phdd_if_macro_internal:n #1 { p , T , F , TF }
  {
    \bool_if:NTF \l__phdd_macro_internal_set_bool
      {
        \bool_if:NTF \l__phdd_macro_internal_bool
```

```
          { \prg_return_true: } { \prg_return_false: }
      }
      {
        \tl_if_empty:fTF
          {
            \exp_after:wN \__phdd_if_macro_internal_aux:w
            \tl_to_str:n { #1 ~ __ }
          }
          { \prg_return_false: } { \prg_return_true: }
      }
  }
\exp_last_unbraced:NNNNo
  \cs_new:Npn \__phdd_if_macro_internal_aux:w #1 { \tl_to_str:n { __ } } { }
```

(*End definition for* `\__phdd_if_macro_internal:nTF` *and* `\__phdd_if_macro_internal_aux:w`.)

`mes_block_base_map:N`

The **`\l__phdd_names_block_tl`** contains sequence variables corresponding to different base functions and their variants. For each such sequence, put the first and second items in **`\l__phdd_tmpa_tl`** and **`\l__phdd_tmpb_tl`** and build the base function's name.

```
\cs_new_protected:Npn \__phdd_names_block_base_map:N #1
  {
    \tl_map_inline:Nn \l__phdd_names_block_tl
      {
        \group_begin:
          \seq_set_eq:NN \l__phdd_tmpa_seq ##1
          \seq_pop:NN \l__phdd_tmpa_seq \l__phdd_tmpa_tl
          \seq_get:NN \l__phdd_tmpa_seq \l__phdd_tmpb_tl
          \exp_args:NNx
        \group_end:
        #1
          {
            \l__phdd_tmpa_tl
            \str_if_eq:VnF \l__phdd_tmpb_tl { \scan_stop: }
              { : \l__phdd_tmpb_tl }
            \bool_if:NT \l__phdd_macro_TF_bool { TF }
          }
      }
  }
```

(*End definition for* `\__phdd_names_block_base_map:N`.)

D.2.40 The `function` environment

```
\keys_define:nn { l3doc/function }
  {
    TF .value_forbidden:n = true ,
    TF .code:n =
      {
        \bool_set_true:N \l__phdd_macro_TF_bool
      } ,
    EXP .value_forbidden:n = true ,
    EXP .code:n =
      {
```

```
        \bool_set_true:N \l__phdd_macro_EXP_bool
        \bool_set_false:N \l__phdd_macro_rEXP_bool
      } ,
    rEXP .value_forbidden:n = true ,
    rEXP .code:n =
      {
        \bool_set_false:N \l__phdd_macro_EXP_bool
        \bool_set_true:N \l__phdd_macro_rEXP_bool
      } ,
    pTF .value_forbidden:n = true ,
    pTF .code:n =
      {
        \bool_set_true:N \l__phdd_macro_pTF_bool
        \bool_set_true:N \l__phdd_macro_TF_bool
        \bool_set_true:N \l__phdd_macro_EXP_bool
        \bool_set_false:N \l__phdd_macro_rEXP_bool
      } ,
    noTF .value_forbidden:n = true ,
    noTF .code:n =
      {
        \bool_set_true:N \l__phdd_macro_noTF_bool
        \bool_set_true:N \l__phdd_macro_TF_bool
      } ,
    added .code:n = { \__phdd_date_set:Nn \l__phdd_date_added_tl {#1} },
    updated .code:n = { \__phdd_date_set:Nn \l__phdd_date_updated_tl {#1} } ,
    deprecated .code:n = { \__phdd_deprecated_on:n {#1} } ,
    tested .code:n = { } ,
    label .code:n =
      {
        \clist_set:Nn \l__phdd_function_label_clist {#1}
        \bool_set_true:N \l__phdd_no_label_bool
      } ,
    verb .value_forbidden:n = true ,
    verb .bool_set:N = \l__phdd_names_verb_bool ,
    module .tl_set:N = \l__phdd_override_module_tl ,
  }
```

`\__phdd_date_set:Nn` Normalize the date into the format `YYYY-MM-DD`; more precisely month and day are allowed to be single digits.

```
\cs_new_protected:Npn \__phdd_date_set:Nn #1#2
  {
    \tl_set:Nn #1 {#2}
    \regex_replace_once:nnNF
      { \A(\d\d\d\d)[-/](\d\d?)[-/](\d\d?)\Z } { \1-\2-\3 } #1
      {
        \msg_error:nnn { phdd } { date-format } {#2}
        \tl_set:Nn #1 { 1970-01-01 }
      }
  }
```

(*End definition for* `\__phdd_date_set:Nn`.)

`phdd_deprecated_on:n` The date comparison function expects two dates in the YYYY-MM-DD format (- is not subtraction here). Complain if a deprecated function should have been

removed earlier. In any case, mark it as internal to suppress the text "documented on page ...".

```
\cs_new_protected:Npn \__phdd_deprecated_on:n #1
  {
    \__phdd_date_set:Nn \l__phdd_tmpa_tl {#1}
    \exp_args:No \__phdd_date_compare:nNnT
      { \l__phdd_tmpa_tl } < { \tex_year:D - \tex_month:D - \tex_day:D }
      {
        \msg_error:nnxx { l3doc } { deprecated-function }
          { \tl_to_str:N \l__phdd_macro_argument_tl }
          { \l__phdd_tmpa_tl }
      }
    \bool_set_true:N \l__phdd_macro_internal_bool
    \bool_set_true:N \l__phdd_macro_internal_set_bool
  }
```

(*End definition for* `\__phdd_deprecated_on:n`.)

`\__phdd_function:nnw`

1. Key–value list.
2. Comma-separated list of functions; input has already been sanitised by catcode changes before reading the argument.

`__phdd_function_end:`

Make sure any paragraph is finished, and similar safe practices at the beginning of an environment which will typeset material. Initialize some variables. Parse the key–value list. Clean up the list of functions, then go through them to extract some data. After this, typeset the function names in the coffin `\l__phdd_functions_coffin` and measure it to know if it fits in the margin. Finally, start a vertical coffin for the main part of the environment. This coffin stops when the environment ends, then all the pieces are assembled into a single coffin, which is typeset.

```
\cs_new_protected:Npn \__phdd_function:nnw #1#2
  {
    \__phdd_function_typeset_start:
    \__phdd_function_init:
    \tl_set:Nn \l__phdd_macro_argument_tl {#2}
    \keys_set:nn { l3doc/function } {#1}
    \__phdd_names_get_seq:nN {#2} \l__phdd_names_seq
    \__phdd_names_parse:
    \__phdd_function_typeset:
    \__phdd_function_reset:
    \__phdd_function_descr_start:w
  }
\cs_new_protected:Npn \__phdd_function_end:
  {
    \__phdd_function_descr_stop:
    \__phdd_function_assemble:
    \__phdd_function_typeset_stop:
  }
```

(*End definition for* `\__phdd_function:nnw` *and* `\__phdd_function_end:`.)

`ction_typeset_start:`
`nction_typeset_stop:`

At the start of the `function` environment, before performing any assignment, close the last paragraph, and set up the typesetting scene. Further code typesets a coffin, so we end the paragraph and allow a page break.

```
\cs_new_protected:Npn \__phdd_function_typeset_start:
  {
    \par \bigskip \noindent
  }
\cs_new_protected:Npn \__phdd_function_typeset_stop:
  {
    \par
    \allowbreak
  }
```

(*End definition for* `\__phdd_function_typeset_start:` *and* `\__phdd_function_typeset_stop:`.)

`_phdd_function_init:` Complain if `function` environments are nested. Clear various variables.

```
\cs_new_protected:Npn \__phdd_function_init:
  {
    \box_if_empty:NF \g__phdd_syntax_box
      { \msg_error:nn { l3doc } { syntax-nested-function } }
    \coffin_clear:N \l__phdd_descr_coffin
    \box_gclear:N \g__phdd_syntax_box
    \coffin_clear:N \l__phdd_syntax_coffin
    \coffin_clear:N \l__phdd_functions_coffin
    \bool_set_false:N \l__phdd_macro_TF_bool
    \bool_set_false:N \l__phdd_macro_pTF_bool
    \bool_set_false:N \l__phdd_macro_noTF_bool
    \bool_set_false:N \l__phdd_macro_EXP_bool
    \bool_set_false:N \l__phdd_macro_rEXP_bool
    \bool_set_false:N \l__phdd_no_label_bool
    \bool_set_false:N \l__phdd_names_verb_bool
    \bool_set_true:N \l__phdd_in_function_bool
    \clist_clear:N \l__phdd_function_label_clist
    \tl_set:Nn \l__phdd_override_module_tl { \q_no_value }
    \char_set_active_eq:NN \< \__phdd_shorthand_meta:
    \char_set_catcode_active:N \<
  }
```

(*End definition for* `\__phdd_function_init:`.)

`phdd_shorthand_meta:` `hdd_shorthand_meta:w` Allow `<...>` to be used as markup for `\meta{...}`. As for example {⟨*test*⟩}

```
\cs_new_protected:Npn \__phdd_shorthand_meta:
  { \mode_if_math:TF { < } { \__phdd_shorthand_meta:w } }

\cs_new_protected_nopar:Npn \__phdd_shorthand_meta:w #1 > { \meta {#1} }
```

(*End definition for* `\__phdd_shorthand_meta:` *and* `\__phdd_shorthand_meta:w`.)

`phdd_function_reset:` Clear some variables.

```
\cs_new_protected:Npn \__phdd_function_reset:
  {
    \tl_set:Nn \l__phdd_override_module_tl { \q_no_value }
  }
```

(*End definition for* `\__phdd_function_reset:`.)

`dd_function_typeset:` Typeset in the coffin `\l__phdd_functions_coffin` the functions listed in `\l__phdd_names_block_tl` and the relevant dates, then set `\l__phdd_long_name_bool` to be `true` if this coffin is larger than the available width in

the margin. The function **__phdd_typeset_functions:** is quite involved hence given later.

```
\cs_new_protected:Npn \__phdd_function_typeset:
  {
    \dim_zero:N \l__phdd_trial_width_dim
    \hcoffin_set:Nn \l__phdd_functions_coffin { \__phdd_typeset_functions: }
    \dim_set:Nn \l__phdd_trial_width_dim
      { \box_wd:N \l__phdd_functions_coffin }
    \bool_set:Nn \l__phdd_long_name_bool
      { \dim_compare_p:nNn \l__phdd_trial_width_dim > \marginparwidth }
  }
```

(*End definition for* `\__phdd_function_typeset:`.)

nction_descr_start:w
function_descr_stop:

The last step in **__phdd_function:nnw** (the beginning of a `function` environment) is to open a coffin which will capture the description of the function, namely the body of the `function` environment. This is closed by **__phdd_function_end:** (the end of a `function` environment).

```
\cs_new_protected:Npn \__phdd_function_descr_start:w
  {
    \vcoffin_set:Nnw \l__phdd_descr_coffin { \textwidth }
      \noindent \ignorespaces
  }
\cs_new_protected:Npn \__phdd_function_descr_stop:
  { \vcoffin_set_end: }
```

(*End definition for* `\__phdd_function_descr_start:w` *and* `\__phdd_function_descr_stop:`.)

d_function_assemble:

The box **\g__phdd_syntax_box** contains the contents of the syntax environment if it was used. Now that we have all the pieces, join together the syntax coffin, the names coffin, and the description coffin. The relative positions depend on whether the names coffin fits in the margin. Then typeset the combination.

```
\cs_new_protected:Npn \__phdd_function_assemble:
  {
    \hcoffin_set:Nn  \l__phdd_syntax_coffin
      { \box_use_drop:N \g__phdd_syntax_box }
    \bool_if:NTF \l__phdd_long_name_bool
      {
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {hc} {vc}
          \l__phdd_syntax_coffin {l} {T}
          {0pt} {0pt}
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {l} {t}
          \l__phdd_functions_coffin  {r} {t}
          {-\marginparsep} {0pt}
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {l} {b}
          \l__phdd_descr_coffin  {l} {t}
          {0.75\marginparwidth + \marginparsep} {-
\medskipamount}
        \coffin_typeset:Nnnnn \l__phdd_output_coffin
          {\l__phdd_descr_coffin-l} {\l__phdd_descr_coffin-t}
          {0pt} {0pt}
```

```
      }
      {
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {hc} {vc}
          \l__phdd_syntax_coffin {l} {t}
          {0pt} {0pt}
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {l} {b}
          \l__phdd_descr_coffin  {l} {t}
          {0pt} {-\medskipamount}
        \coffin_join:NnnNnnnn
          \l__phdd_output_coffin {l} {t}
          \l__phdd_functions_coffin  {r} {t}
          {-\marginparsep} {0pt}
        \coffin_typeset:Nnnnn \l__phdd_output_coffin
          {\l__phdd_syntax_coffin-l} {\l__phdd_syntax_coffin-T}
          {0pt} {0pt}
      }
  }
```

(*End definition for* `\__phdd_function_assemble:`.)

`d_typeset_functions:`

This function builds the **`\l__phdd_functions_coffin`** by typesetting the function names (with variants) and the relevant dates in a `tabular` environment. The use of rules **`\toprule`**, **`\midrule`** and **`\bottomrule`** requires whatever lies between the last **`\\`** and the rule to be expandable, making our lives a bit complicated.

```
\cs_new_protected:Npn \__phdd_typeset_functions:
  {
    \small\ttfamily
    \HD@savedestfalse
    \HD@target
    \Hy@MakeCurrentHref { HD. \int_use:N \c@HD@hypercount }
    \renewcommand{\arraystretch}{1.6}
    \begin{tabular} [b] { @{} l @{} >{\hspace{\tabcolsep}} r @{} }
      \hline
      \__phdd_function_extra_labels:
      \__phdd_names_typeset:
      \__phdd_typeset_dates:
      \hline
    \end{tabular}
    \normalfont\normalsize
  }
```

(*End definition for* `\__phdd_typeset_functions:`.)

`et_function_block:nN`
`et_function_block:xN`
`hdd_function_index:n`
`hdd_function_index:x`

#1 is a csname, **#2** a boolean indicating whether to add **TF** or not.

```
\cs_new_protected:Npn \__phdd_typeset_function_block:nN #1#2
  {
    \__phdd_function_index:x
      { #1 \bool_if:NT #2 { \tl_to_str:n {TF} } }
    \__phdd_function_label:xN {#1} #2
    #1
    \bool_if:NT #2 { \__phdd_typeset_TF: }
```

```
    \__phdd_typeset_expandability:
    \seq_if_empty:NF \g__phdd_variants_seq
      { \__phdd_typeset_variant_list:nN {#1} #2 }
    \\
  }
\cs_generate_variant:Nn \__phdd_typeset_function_block:nN { x }
\cs_new_protected:Npn \__phdd_function_index:n #1
  {
    \seq_gput_right:Nn \g_doc_functions_seq {#1}
    \__phdd_special_index:nn {#1} { usage }
  }
\cs_generate_variant:Nn \__phdd_function_index:n { x }
\cs_new_protected:Npn \__phdd_typeset_expandability:
  {
    &
    \bool_if:NT \l__phdd_macro_EXP_bool  { \__phdd_typeset_exp: }
    \bool_if:NT \l__phdd_macro_rEXP_bool { \__phdd_typeset_rexp: }
  }
```

`#1` is the function, `#2` whether to add `TF`.

```
\cs_new_protected:Npn \__phdd_typeset_variant_list:nN #1#2
  {
    \\
    \__phdd_typeset_aux:n { \__phdd_get_function_name:n {#1} }
    :
    \int_compare:nTF { \seq_count:N \g__phdd_variants_seq == 1 }
      { \seq_use:Nn \g__phdd_variants_seq { } }
      {
        \textrm(
          \seq_use:Nn \g__phdd_variants_seq { \textrm| }
        \textrm)
      }
    \bool_if:NT #2 { \__phdd_typeset_TF: }
    \__phdd_typeset_expandability:
  }
```

`#1` is the function name, `#2` whether to add `TF`.

```
\cs_new_protected:Npn \__phdd_function_extra_labels:
  {
    \bool_if:NT \l__phdd_no_label_bool
      {
        \clist_map_inline:Nn \l__phdd_function_label_clist
          {
            \__phdd_get_hyper_target:oN { \token_to_str:N ##1 }
              \l__phdd_tmpa_tl
            \exp_args:No \label { \l__phdd_tmpa_tl }
          }
      }
  }
\cs_new_protected:Npn \__phdd_function_label:nN #1#2
  {
    \bool_if:NF \l__phdd_no_label_bool
      {
        \__phdd_get_hyper_target:xN
          {
```

```
            \exp_not:n {#1}
            \bool_if:NT #2 { \tl_to_str:n {TF} }
          }
          \l__phdd_tmpa_tl
        \exp_args:No \label { \l__phdd_tmpa_tl }
      }
  }
\cs_generate_variant:Nn \__phdd_function_label:nN { x }
```

(*End definition for* `\__phdd_typeset_function_block:nN` *and* `\__phdd_function_index:n`.)

`\__phdd_typeset_dates:` To display metadata for when functions are added/modified. This function must be expandable since it produces rules for use in alignments.

```
\cs_new:Npn \__phdd_typeset_dates:
  {
    \bool_lazy_and:nnF
      { \tl_if_empty_p:N \l__phdd_date_added_tl }
      { \tl_if_empty_p:N \l__phdd_date_updated_tl }
      { \midrule }
    \tl_if_empty:NF \l__phdd_date_added_tl
      {
        \multicolumn { 2 } { @{} r @{} }
          { \scriptsize New: \, \l__phdd_date_added_tl } \\
      }

    \tl_if_empty:NF \l__phdd_date_updated_tl
      {
        \multicolumn { 2 } { @{} r @{} }
          { \scriptsize Updated: \, \l__phdd_date_updated_tl } \\
      }
  }
```

(*End definition for* `\__phdd_typeset_dates:`.)

`\__phdd_syntax:w` `\__phdd_syntax_end:` Implement the `syntax` environment.

```
\dim_new:N \l__phdd_syntax_dim
\cs_new_protected:Npn \__phdd_syntax:w
  {
    \box_if_empty:NF \g__phdd_syntax_box
      { \msg_error:nn { l3doc } { multiple-syntax } }
    \dim_set:Nn \l__phdd_syntax_dim
      {
        \textwidth
        \bool_if:NT \l__phdd_long_name_bool
          { + 0.75 \marginparwidth - \l__phdd_trial_width_dim }
      }
    \hbox_gset:Nw \g__phdd_syntax_box
      \small \ttfamily
      \arrayrulecolor{white}
      \begin{tabular} { @{} l @{} }
        \toprule
        \begin{minipage}[t]{\l__phdd_syntax_dim}
          \raggedright
          \obeyspaces
          \obeylines
```

```
  }
\cs_new_protected:Npn \__phdd_syntax_end:
  {
        \end{minipage}
      \end{tabular}
      \arrayrulecolor{black}
    \hbox_gset_end:
    \bool_if:NF \l__phdd_in_function_bool
      {
        \begin{quote}
          \mode_leave_vertical:
          \box_use_drop:N \g__phdd_syntax_box
        \end{quote}
      }
  }
```

(*End definition for* `\__phdd_syntax:w` *and* `\__phdd_syntax_end:`.)

D.2.40 The `macro` environment

Keyval for the `macro` environment. TODO: provide document command for documenting keys.

```
\keys_define:nn { phdd/macro }
  {
    aux .value_forbidden:n = true ,
    aux .code:n =
      {
        \msg_warning:nnnn { phdd } { deprecated-option }
          { aux } { function/macro }
      } ,
    internal .value_forbidden:n = true ,
    internal .code:n =
      {
        \bool_set_true:N \l__phdd_macro_internal_bool
        \bool_set_true:N \l__phdd_macro_internal_set_bool
      } ,
    int .value_forbidden:n = true ,
    int .code:n =
      {
        \bool_set_true:N \l__phdd_macro_internal_bool
        \bool_set_true:N \l__phdd_macro_internal_set_bool
      } ,
    var .value_forbidden:n = true ,
    var .code:n =
      { \bool_set_true:N \l__phdd_macro_var_bool } ,
    TF .value_forbidden:n = true ,
    TF .code:n =
      { \bool_set_true:N \l__phdd_macro_TF_bool } ,
    pTF .value_forbidden:n = true ,
    pTF .code:n =
      {
        \bool_set_true:N \l__phdd_macro_TF_bool
        \bool_set_true:N \l__phdd_macro_pTF_bool
```

```
        \bool_set_true:N \l__phdd_macro_EXP_bool
        \bool_set_false:N \l__phdd_macro_rEXP_bool
      } ,
    noTF .value_forbidden:n = true ,
    noTF .code:n =
      {
        \bool_set_true:N \l__phdd_macro_TF_bool
        \bool_set_true:N \l__phdd_macro_noTF_bool
      } ,
    EXP .value_forbidden:n = true ,
    EXP .code:n =
      {
        \bool_set_true:N \l__phdd_macro_EXP_bool
        \bool_set_false:N \l__phdd_macro_rEXP_bool
      } ,
    rEXP .value_forbidden:n = true ,
    rEXP .code:n =
      {
        \bool_set_false:N \l__phdd_macro_EXP_bool
        \bool_set_true:N \l__phdd_macro_rEXP_bool
      } ,
    tested .code:n =
      {
        \bool_set_true:N \l__phdd_macro_tested_bool
      } ,
    added .code:n = {} , % TODO
    updated .code:n = {} , % TODO
    deprecated .code:n = { \__phdd_deprecated_on:n {#1} } ,
    verb .bool_set:N = \l__phdd_names_verb_bool ,
    module .tl_set:N = \l__phdd_override_module_tl ,
    documented-as .tl_set:N = \l__phdd_macro_documented_tl ,
  }
```

`\__phdd_macro:nnw` The arguments are a key–value list of ⟨*options*⟩ and a comma-list of ⟨*names*⟩, read verbatim by **xparse**. First initialize some variables before applying the ⟨*options*⟩, then parse the ⟨*names*⟩ to get a sequence of macro names, then apply `\__phdd_macro_single:nNN` to each (this step is more subtle than `\seq_map_function:NN` because of `TF`/`pTF`/`noTF`). Finally typeset the macro names in the margin.

```
\cs_new_protected:Npn \__phdd_macro:nnw #1#2
  {
    \__phdd_macro_init:
    \tl_set:Nn \l__phdd_macro_argument_tl {#2}
    \keys_set:nn { phdd/macro } {#1}
    \__phdd_names_get_seq:nN {#2} \l__phdd_names_seq
    \__phdd_names_parse:
    \__phdd_macro_save_names:
    \__phdd_names_typeset:
    \__phdd_macro_dump:
    \__phdd_macro_reset:
  }
```

(*End definition for* `\__phdd_macro:nnw`.)

`\__phdd_macro_init:` The booleans hold various key–value options, `\l__phdd_nested_macro_-`

int counts the number of `macro` environments around the current point (is 0 outside).

```
\cs_new_protected:Npn \__phdd_macro_init:
  {
    \int_incr:N \l__phdd_nested_macro_int
    \bool_set_false:N \l__phdd_macro_internal_bool
    \bool_set_false:N \l__phdd_macro_internal_set_bool
    \bool_set_false:N \l__phdd_macro_TF_bool
    \bool_set_false:N \l__phdd_macro_pTF_bool
    \bool_set_false:N \l__phdd_macro_noTF_bool
    \bool_set_false:N \l__phdd_macro_EXP_bool
    \bool_set_false:N \l__phdd_macro_rEXP_bool
    \bool_set_false:N \l__phdd_macro_var_bool
    \bool_set_false:N \l__phdd_macro_tested_bool
    \bool_set_false:N \l__phdd_names_verb_bool
    \tl_set:Nn \l__phdd_override_module_tl { \q_no_value }
    \tl_clear:N \l__phdd_macro_documented_tl
    \cs_set_eq:NN \testfile \__phdd_print_testfile:n
    \box_clear:N \l__phdd_macro_index_box
    \vbox_set:Nn \l__phdd_macro_box
      {
        \hbox:n
          {
            \strut
            \int_compare:nNnT \l__phdd_macro_int = 0
              { \HD@target }
          }
        \vskip \int_eval:n { \l__phdd_macro_int - 1 } \baselineskip
      }
  }
```

(*End definition for* `\__phdd_macro_init:`.)

`\__phdd_macro_reset:` We ensure that **\cs** commands nested inside a macro whose module is imposed are not affected.

```
\cs_new_protected:Npn \__phdd_macro_reset:
  {
    \tl_set:Nn \l__phdd_override_module_tl { \q_no_value }
  }
```

(*End definition for* `\__phdd_macro_reset:`.)

`dd_macro_save_names:` The list of names defined in a set of `macro` environments is eventually used to display on which page they are documented. If the **documented-as** key is given, use that, otherwise find names in **\l__phdd_names_block_tl**.

```
\cs_new_protected:Npn \__phdd_macro_save_names:
  {
    \tl_if_empty:NTF \l__phdd_macro_documented_tl
      { \__phdd_names_block_base_map:N \__phdd_macro_save_names_aux:n }
      {
        \seq_gput_right:Nf \g__phdd_nested_names_seq
          { \exp_after:wN \token_to_str:N \l__phdd_macro_documented_tl }
      }
  }
```

```
\cs_new_protected:Npn \__phdd_macro_save_names_aux:n #1
  { \seq_gput_right:Nn \g__phdd_nested_names_seq {#1} }
```

(*End definition for* `\__phdd_macro_save_names:`.)

`\__phdd_macro_dump:` This calls `\makelabel{}`

```
\cs_new_protected:Npn \__phdd_macro_dump:
  {
    \topsep\MacroTopsep
    \trivlist
    \cs_set:Npn \makelabel ##1
      {
        \llap
          {
            \hbox_unpack_clear:N \l__phdd_macro_index_box
            \vtop to \baselineskip
              {
                \vbox_unpack_clear:N \l__phdd_macro_box
                \vss
              }
          }
      }
    \item [ ]
  }
```

(*End definition for* `\__phdd_macro_dump:`.)

`cro_typeset_block:nN` Used to typeset a macro and its variants. `#1` is the macro name, `#2` is a boolean controlling whether to add `TF`.

```
\cs_new_protected:Npn \__phdd_macro_typeset_block:nN #1#2
  {
    \__phdd_macro_single:nNN {#1} \c_true_bool #2
    \seq_if_empty:NF \g__phdd_variants_seq
      {
        \__phdd_macro_typeset_variant_list:xN
          { \__phdd_get_function_name:n {#1} } #2
      }
  }
\cs_generate_variant:Nn \__phdd_macro_typeset_block:nN { x }
\cs_new_protected:Npn \__phdd_macro_typeset_variant_list:nN #1#2
  {
    \seq_map_inline:Nn \g__phdd_variants_seq
      { \__phdd_macro_single:nNN { #1 : ##1 } \c_false_bool #2 }
  }
\cs_generate_variant:Nn \__phdd_macro_typeset_variant_list:nN { x }
```

(*End definition for* `\__phdd_macro_typeset_block:nN`.)

`hdd_macro_single:nNN` The arguments are `#1` a macro name (without `TF`), `#2` a boolean determining whether or not to index, and `#3` whether or not to add `TF`. Let's start to mess around with doc's `macro` environment. See `doc.dtx` for a full explanation of the original environment. It's rather *enthusiastically* commented.

1. Macro/function/whatever name; input has already been sanitised.

The assignments to **\saved@macroname** and **\saved@indexname** are used by **doc**'s **\changes** mechanism.

```
\cs_new_protected:Npn \__phdd_macro_single:nNN #1#2#3
  {
    \tl_set:Nn \saved@macroname {#1}
    \__phdd_macro_typeset_one:nN {#1} #3
    \bool_if:NT #3 { \DoNotIndex {#1} }
    \exp_args:Nx \__phdd_macro_index:nN
      { #1 \bool_if:NT #3 { \tl_to_str:n { TF } } }
      #2
  }
\cs_new_protected:Npn \__phdd_macro_index:nN #1#2
  {
    \DoNotIndex {#1}
    \bool_if:NT #2
      {
        \__phdd_if_macro_internal:nF {#1}
          { \seq_gput_right:Nn \g_doc_macros_seq {#1} }
        \hbox_set:Nw \l__phdd_macro_index_box
          \hbox_unpack_clear:N \l__phdd_macro_index_box
          \int_gincr:N \c@CodelineNo
          \__phdd_special_index:nn {#1} { main }
          \int_gdecr:N \c@CodelineNo
        \exp_args:NNNo \hbox_set_end:
          \tl_set:Nn \saved@indexname { \l__phdd_index_key_tl }
      }
  }
```

(*End definition for* `\__phdd_macro_single:nNN`.)

macro_typeset_one:nN

For a long time, l3doc collected the macro names as labels in the first items of nested **\trivlist**, but these were not closed properly with **\endtrivlist**. Also, it interacted in surprising ways with **hyperref** targets. Now, we collect typeset macro names by hand in the box **\l__phdd_macro_box**. Note the space **\ . #1** is the macro name, **#2** whether to add **TF**. TODO see if we need to label it here.

```
\cs_new_protected:Npn \__phdd_macro_typeset_one:nN #1#2
  {
    \vbox_set:Nn \l__phdd_macro_box
      {
        \vbox_unpack_clear:N \l__phdd_macro_box
        % YL added 15pt if used in conjuction with listings to avoid overlapping
        % still probelms lining up with codeline. Need to hack doc or doc-
v3.
        \hbox { \llap { \__phdd_print_macroname:nN {#1} #2 \space}\space\space\space\spa
      }
    \int_incr:N \l__phdd_macro_int
  }
```

(*End definition for* `\__phdd_macro_typeset_one:nN`.)

d_print_macroname:nN

In the name, spaces are replaced by other spaces to ensure they get displayed in case there are any.

```
\cs_new_protected:Npn \__phdd_print_macroname:nN #1#2
```

```
  {
    \strut
    \__phdd_get_hyper_target:xN
      {
        \exp_not:n {#1}
        \bool_if:NT #2 { \tl_to_str:n {TF} }
      }
      \l__phdd_tmpa_tl
    \cs_if_exist:cTF { r@ \l__phdd_tmpa_tl }
      { \exp_last_unbraced:NNo \hyperref [ \l__phdd_tmpa_tl ] }
      { \use:n }
      {
        \int_compare:nTF { \str_count:n {#1} <= 28 }
          { \MacroFont } {\MacroFont} %{ \MacroLongFont }
        \tl_set:Nn \l__phdd_tmpa_tl {#1}
        \tl_replace_all:Nno \l__phdd_tmpa_tl
          { ~ } { \c_catcode_other_space_tl }
        \__phdd_macroname_prefix:o \l__phdd_tmpa_tl
        \__phdd_macroname_suffix:N #2
      }
  }
\cs_new_protected:Npn \__phdd_macroname_prefix:n #1
  {
    \__phdd_if_macro_internal:nTF {#1}
      { \__phdd_typeset_aux:n {#1} } {#1}
  }
\cs_generate_variant:Nn \__phdd_macroname_prefix:n { o }
\cs_new_protected:Npn \__phdd_macroname_suffix:N #1
  { \bool_if:NTF #1 { \__phdd_typeset_TF: } { } }
```

(*End definition for* `\__phdd_print_macroname:nN`.)

`\MacroLongFont`

```
\providecommand \MacroLongFont
  {
    \fontfamily{lmtt}\fontseries{lc}\small
  }
```

(*End definition for* `\MacroLongFont`.)

`\__phdd_print_testfile:n`
`\__phdd_print_testfile_aux:n`

Used to show that a macro has a test, somewhere.

```
\cs_new_protected:Npn \__phdd_print_testfile:n #1
  {
    \bool_set_true:N \l__phdd_macro_tested_bool
    \tl_if_eq:nnF {#1} {*}
      {
        \seq_if_in:NnF \g__phdd_testfiles_seq {#1}
          {
            \seq_gput_right:Nn \g__phdd_testfiles_seq {#1}
            \par
            \__phdd_print_testfile_aux:n {#1}
          }
      }
  }
\cs_new_protected:Npn \__phdd_print_testfile_aux:n #1
```

```
  {
    \footnotesize
    (
    \textit
      {
        The~ test~ suite~ for~ this~ command,~
        and~ others~ in~ this~ file,~ is~ \textsf{#1}
      }.
    )\par
  }
```

(*End definition for* `\__phdd_print_testfile:n` *and* `\__phdd_print_testfile_aux:n`.)

`\TestFiles`

```
\DeclareDocumentCommand \TestFiles {m}
  {
    \par
    \textit
      {
        The~ following~ test~ files~ are~
        used~ for~ this~ code:~ \textsf{#1}.
      }
    \par \ignorespaces
  }
```

(*End definition for* `\TestFiles`.)

`\UnitTested`

```
\DeclareDocumentCommand \UnitTested { } { \testfile* }
```

(*End definition for* `\UnitTested`.)

`\TestMissing`

```
\DeclareDocumentCommand \TestMissing { m }
  { \__phdd_test_missing:n {#1} }
```

(*End definition for* `\TestMissing`.)

`_phdd_test_missing:n` Keys in `\g__phdd_missing_tests_prop` are lists of macros given as arguments of one `macro` environment. Values are pairs of a file name and a comment about the missing tests.

```
\cs_new_protected:Npn \__phdd_test_missing:n #1
  {
    \__phdd_test_missing_aux:Nxn
      \g__phdd_missing_tests_prop
      { \seq_use:Nn \l__phdd_names_seq { , } }
      { { \g_file_curr_name_str \c_space_tl (#1) } }
  }
\cs_new_protected:Npn \__phdd_test_missing_aux:Nnn #1#2#3
  {
    \prop_get:NnNTF #1 {#2} \l__phdd_tmpa_tl
      { \tl_put_right:Nn \l__phdd_tmpa_tl { , #3 } }
      { \tl_set:Nn \l__phdd_tmpa_tl {#3} }
    \prop_put:Nno #1 {#2} \l__phdd_tmpa_tl
  }
\cs_generate_variant:Nn \__phdd_test_missing_aux:Nnn { Nx }
```

(*End definition for* `\__phdd_test_missing:n`.)

`\__phdd_macro_end:` It is too late for anyone to declare a test file for this macro, so we can check now whether the macro is tested. If the `macro` environment which is being ended is the outermost one, then wrap each macro in **`\texttt`** (with the addition of **`TF`** if relevant) and typeset two informations: that this ends the definition of some macros, and that they are documented on some page.

```
\cs_new_protected:Npn \__phdd_macro_end:
  {
    \endtrivlist
    \__phdd_macro_end_check_tested:
    \int_compare:nNnT \l__phdd_nested_macro_int = 1
      { \__phdd_macro_end_style:n { \__phdd_print_end_definition: } }
  }
```

(*End definition for* `\__phdd_macro_end:`.)

`ro_end_check_tested:` If the **`checktest`** option was issued and the macro is not an auxiliary nor a variable (and it does not have a test), then add it to the sequence of non-tested macros.

```
\cs_new_protected:Npn \__phdd_macro_end_check_tested:
  {
    \bool_lazy_all:nT
     {
       { \g__phdd_checktest_bool }
       { ! \l__phdd_macro_var_bool }
       { ! \l__phdd_macro_tested_bool }
     }
     {
       \seq_set_filter:NNn \l__phdd_tmpa_seq \l__phdd_names_seq
         { ! \__phdd_if_macro_internal_p:n {##1} }
       \seq_gput_right:Nx \g__phdd_not_tested_seq
         {
           \seq_use:Nn \l__phdd_tmpa_seq { , }
           \bool_if:NTF \l__phdd_macro_pTF_bool {~(pTF)}
             { \bool_if:NT \l__phdd_macro_TF_bool {~(TF)} }
         }
     }
  }
```

(*End definition for* `\__phdd_macro_end_check_tested:`.)

`dd_macro_end_style:n` Style for the extra information at the end of a top-level `macro` environment.

```
\cs_new_protected:Npn \__phdd_macro_end_style:n #1
  {
    \nobreak \noindent
    { \footnotesize ( \emph{#1} ) \par }
  }
```

(*End definition for* `\__phdd_macro_end_style:n`.)

`rint_end_definition:`
`acro_end_wrap_item:n`
`dd_print_documented:` Surround each item by **`\texttt`**, replacing **`_`** by **`\_`** as well. Then list the macro names through **`\seq_use:Nnnn`**, unless there are too many. Finally, if the macro is neither auxiliary nor internal, add a link to where it is documented.

```
\cs_new_protected:Npn \__phdd_macro_end_wrap_item:n #1
  {
    \tl_set:Nn \l__phdd_tmpa_tl {#1}
    \tl_replace_all:Non \l__phdd_tmpa_tl
      { \token_to_str:N _ } { \_ }
    \texttt { \l__phdd_tmpa_tl }
  }
\cs_new_protected:Npn \__phdd_print_end_definition:
  {
    \seq_set_map:NNn \l__phdd_tmpa_seq
      \g__phdd_nested_names_seq
      { \exp_not:n { \__phdd_macro_end_wrap_item:n {##1} } }
    End~ definition~ for~
    \int_compare:nTF { \seq_count:N \l__phdd_tmpa_seq <= 3 }
      {
        \seq_use:Nnnn \l__phdd_tmpa_seq
          { \,~and~ } { \,,~ } { \,,~and~ }
      }
      { \seq_item:Nn \l__phdd_tmpa_seq {1}\,~and~others }
    \@.
    \__phdd_print_documented:
  }
\cs_new_protected:Npn \__phdd_print_documented:
  {
    \seq_gset_filter:NNn \g__phdd_nested_names_seq
      \g__phdd_nested_names_seq
      { ! \__phdd_if_macro_internal_p:n {##1} }

    \seq_if_empty:NF \g__phdd_nested_names_seq
      {
        \int_set:Nn \l__phdd_tmpa_int
          { \seq_count:N \g__phdd_nested_names_seq }
        %\int_compare:nNnTF \l__phdd_tmpa_int = 1 {~This~} {~These~}
        %\bool_if:NTF \l__phdd_macro_var_bool {variable} {function}
        % add switch for this
%        \int_compare:nNnTF \l__phdd_tmpa_int = 1 {~is~} {s~are~}
%        documented~on~page~
%        \__phdd_get_hyper_target:xN
%          { \seq_item:Nn \g__phdd_nested_names_seq { 1 } }
%          \l__phdd_tmpa_tl
%        \exp_args:Nx \pageref { \l__phdd_tmpa_tl } .

      }
    \seq_gclear:N \g__phdd_nested_names_seq
  }
```

(*End definition for* `\__phdd_print_end_definition:`, `\__phdd_macro_end_wrap_item:n`, *and* `\__phdd_print_documented:`.)

`\DescribeOption` For describing package options. Due to Joseph Wright. Name/usage might change soon.

```
\newcommand*{\DescribeOption}
  {
    \leavevmode
    \@bsphack
```

```
    \begingroup
      \MakePrivateLetters
      \Describe@Option
  }
\newcommand*{\Describe@Option}[1]
  {
    \endgroup
    \marginpar{
      \raggedleft
      \PrintDescribeEnv{#1}
    }
    \SpecialOptionIndex{#1}
    \@esphack
    \ignorespaces
  }
\newcommand*{\SpecialOptionIndex}[1]
  {
    \@bsphack
    \begingroup
      \HD@target
      \let\HDorg@encapchar\encapchar
      \edef\encapchar usage
        {
          \HDorg@encapchar hdclindex{\the\c@HD@hypercount}{usage}
        }
      \index
        {
          #1\actualchar{\protect\ttfamily#1}~(option)
          \encapchar usage
        }
      \index
        {
          options:\levelchar#1\actualchar{\protect\ttfamily#1}
          \encapchar usage
        }
    \endgroup
    \@esphack
  }
```

(*End definition for* `\DescribeOption`.)

Here are some definitions for additional markup that helps to structure your documentation.

Provides a danger bend, as known from the TEXbook.

The actual character from the font **manfnt**:

```
\font \manual = manfnt \scan_stop:
\cs_gset:Npn \dbend { {\manual\char127} }
```

Defines the single danger bend. Use it whenever there is a feature in your package that might be tricky to use. FIXME: Has to be fixed when in combination with a macro-definition.

```
%\newenvironment {danger}
%  {
%    \begin{trivlist}\item[]\noindent
```

```
%     \begingroup\hangindent=2pc\hangafter=-2
%     \cs_set:Npn \par{\endgraf\endgroup}
%     \hbox to0pt{\hskip-\hangindent\dbend\hfill}\ignorespaces
%   }
%   {
%     \par\end{trivlist}
%   }
\newenvironment{TemplateInterfaceDescription}[1]
  {
    \subsection{The~object~type~`#1'}
    \begingroup
    \@beginparpenalty\@M
    \description
    \def\TemplateArgument##1##2{\item[Arg:~##1]##2\par}
    \def\TemplateSemantics
      {
        \enddescription\endgroup
        \subsubsection*{Semantics:}
      }
  }
  {
    \par\bigskip
  }
\newenvironment{TemplateDescription}[2]
  {
    \subsection{The~template~`#2'~(object~type~#1)}
    \subsubsection*{Attributes:}
    \begingroup
    \@beginparpenalty\@M
    \description
    \def\TemplateKey##1##2##3##4
      {
        \item[##1~(##2)]##3%
        \ifx\TemplateKey##4\TemplateKey\else
%         \hskip0ptplus3em\penalty-500\hskip 0pt plus 1filll Default:~##4%
          \hfill\penalty500\hbox{}\hfill Default:~##4%
          \nobreak\hskip-\parfillskip\hskip0pt\relax
        \fi
        \par
      }
    \def\TemplateSemantics
      {
        \enddescription\endgroup
        \subsubsection*{Semantics~\&~Comments:}
      }
  }
  { \par \bigskip }
\newenvironment{InstanceDescription}[4][xxxxxxxxxxxxxxx]
  {
    \subsubsection{The~instance~`#3'~(template~#2/#4)}
    \subsubsection*{Attribute~values:}
    \begingroup
    \@beginparpenalty\@M
```

```
    \def\InstanceKey##1##2{\>\textbf{##1}\>##2\\}
    \def\InstanceSemantics{\endtabbing\endgroup
      \vskip-30pt\vskip0pt
      \subsubsection*{Layout~description~\&~Comments:}}
    \tabbing
    xxxx\=#1\=\kill
  }
  { \par \bigskip }
```

D.2.5 Inheriting doc

Code here is taken from **doc**, stripped of comments and translated into **expl3** syntax. New features are added in various places.

`\StopEventually` `\Finale` `\AlsoImplementation` `\OnlyDescription`

TODO: remove these four commands altogether, document that it is better to use the `documentation` and `implementation` environments.

```
\DeclareDocumentCommand \OnlyDescription { }
  { \bool_gset_false:N \g__phdd_typeset_implementation_bool }

\DeclareDocumentCommand \AlsoImplementation { }
  { \bool_gset_true:N \g__phdd_typeset_implementation_bool }

\DeclareDocumentCommand \StopEventually { m }
  {
    \bool_if:NTF \g__phdd_typeset_implementation_bool
      {
        \@bsphack
        \tl_gset:Nn \g__phdd_finale_tl { #1 \check@checksum }
        \init@checksum
        \@esphack
      }
      { #1 \endinput }
  }

\DeclareDocumentCommand \Finale { }
  { \tl_use:N \g__phdd_finale_tl }
\tl_new:N \g__phdd_finale_tl
```

(*End definition for* `\StopEventually` *and others.*)

`\__phdd_input:n`

Inputting a file, with some setup: the module name should be empty before the first `<@@=`⟨*module*⟩`>` line in the file.

```
\cs_new_protected:Npn \__phdd_input:n #1
  {
    \tl_gclear:N \g__phdd_module_name_tl
    \MakePercentIgnore
    \input{#1}
    \MakePercentComment
  }
```

(*End definition for* `\__phdd_input:n`.)

`\DocInput`

Modified from **doc** to accept comma-list input (who has commas in filenames?).

```
\DeclareDocumentCommand \DocInput { m }
  {
    \clist_map_inline:nn {#1}
      {
        \clist_put_right:Nn \g_docinput_clist {##1}
        \__phdd_input:n {##1}
      }
  }
```

(*End definition for* `\DocInput`.)

`\DocInputAgain` Uses `\g_docinput_clist` to re-input whatever's already been `\DocInput`-ed until now. May be used multiple times.

```
\DeclareDocumentCommand \DocInputAgain { }
  { \clist_map_function:NN \g_docinput_clist \__phdd_input:n }
```

(*End definition for* `\DocInputAgain`.)

`\DocInclude` More or less exactly the same as `\include`, but uses `\DocInput` on a `.dtx` file, not `\input` on a `.tex` file.

```
\DeclareDocumentCommand \DocInclude { m }
  {
    \relax\clearpage
    \docincludeaux
    \IfFileExists{#1.fdd}
      { \cs_set:Npn \currentfile{#1.fdd} }
      { \cs_set:Npn \currentfile{#1.dtx} }
    \int_compare:nNnTF \@auxout = \@partaux
      { \@latexerr{\string\include\space cannot~be~nested}\@eha }
      { \@docinclude #1 }
  }
```

```
\cs_gset:Npn \@docinclude #1
  {
    \clearpage
    \immediate\write\@mainaux{\string\@input{#1.aux}}
    \@tempswatrue
    \if@partsw
      \@tempswafalse
      \cs_set:Npx \@tempb{#1}
      \@for\@tempa:=\@partlist\do
        {
          \ifx\@tempa\@tempb\@tempswatrue\fi
        }
    \fi
    \if@tempswa
      \cs_set_eq:NN \@auxout                \@partaux
      \immediate\openout\@partaux #1.aux
      \immediate\write\@partaux{\relax}
      \cs_set_eq:NN \@ltxdoc@PrintIndex     \PrintIndex
      \cs_set_eq:NN \PrintIndex             \relax
      \cs_set_eq:NN \@ltxdoc@PrintChanges   \PrintChanges
      \cs_set_eq:NN \PrintChanges           \relax
      \cs_set_eq:NN \@ltxdoc@theglossary    \theglossary
```

```
      \cs_set_eq:NN \@ltxdoc@endtheglossary  \endtheglossary
      \part{\currentfile}
      {
        \cs_set_eq:NN \ttfamily\relax
        \cs_gset:Npx \filekey
          { \filekey, \thepart = { \ttfamily \currentfile } }
      }
      \DocInput{\currentfile}
      \cs_set_eq:NN \PrintIndex              \@ltxdoc@PrintIndex
      \cs_set_eq:NN \PrintChanges            \@ltxdoc@PrintChanges
      \cs_set_eq:NN \theglossary             \@ltxdoc@theglossary
      \cs_set_eq:NN \endtheglossary          \@ltxdoc@endtheglossary
      \clearpage
      \@writeckpt{#1}
      \immediate \closeout \@partaux
    \else
      \@nameuse{cp@#1}
    \fi
    \cs_set_eq:NN \@auxout \@mainaux
  }
\cs_gset:Npn \codeline@wrindex #1
  {
    \immediate\write\@indexfile
      {
        \string\indexentry{#1}
          { \filesep \int_use:N \c@CodelineNo }
      }
  }
\tl_gclear:N \filesep
```

(*End definition for* `\DocInclude`.)

`\docincludeaux`

```
\cs_gset:Npn \docincludeaux
  {
    \tl_set:Nn \thepart { \alphalph { part } }
    \tl_set:Nn \filesep { \thepart - }
    \cs_set_eq:NN \filekey \use_none:n
    \tl_gput_right:Nn \index@prologue
      {
        \cs_gset:Npn \@oddfoot
          {
            \parbox { \textwidth }
              {
                \strut \footnotesize
                \raggedright { \bfseries File~Key: } ~ \filekey
              }
          }
        \cs_set_eq:NN \@evenfoot \@oddfoot
      }
    \cs_gset_eq:NN \docincludeaux \relax
    \cs_gset:Npn \@oddfoot
      {
        \cs_if_exist:cTF { ver @ \currentfile }
```

```
          { File~\thepart :~{\ttfamily\currentfile}~ }
          {
            \GetFileInfo{\currentfile}
            File~\thepart :~{\ttfamily\filename}~
            Date:~\ExplFileDate\ % space
            Version~\ExplFileVersion
          }
        \hfill \thepage
      }
    \cs_set_eq:NN \@evenfoot \@oddfoot
  }
```

(*End definition for* `\docincludeaux`.)

D.2.50 The `macrocode` environment

`\xmacro@code`
`__phdd_xmacro_code:n`
`__phdd_xmacro_code:w`

Hook into the `macrocode` environment in a dirty way: **`\xmacro@code`** is responsible for grabbing (and tokenizing) the body of the environment. Redefine it to pass what it grabs to **`\__phdd_xmacro_code:n`**. This new macro replaces all `@@` by the appropriate module name. One exceptional case is the `<@@=`⟨*module*⟩`>` lines themselves, where `@@` should not be modified. Actually, we search for such lines, to set the module name automatically. We need to be careful: no `<@@=` should appear as such in the code below since **l3doc** is also typeset using this code.

TODO: right now, in a line containing `<@@=`⟨*module*⟩`>`, the `@@` are replaced (using different values of the ⟨*module*⟩ before and after the assignment). Is this a waste?

```
\RequirePackage{colorize}
```

(*End definition for* `\xmacro@code`, `\__phdd_xmacro_code:n`, *and* `\__phdd_xmacro_code:w`.)

```
/DOCUM
*colorize
\ExplSyntaxOn
\bgroup
\obeylines
\tl_new:N \l__phdd_store_tl
\egroup

\NewDocumentCommand{\codelinefont}{O{blue} m}
  {\begingroup\bfseries
    \color{#1}
    #2
   \endgroup
  }

\def\keyvaluestyle#1
  {
    \begingroup
      =\color{orange}
      %\panunicode #1,
    \endgroup
  }
```

```
\regex_const:Nn \c_ifs_regex { ((\\if(x{1}|num|dimen|cat|eof|false|true|mode|mmode|hmode
         |vmode|\_meaning|_mode_|inner|defined|csname|[@a-zA-
Z]*)*|\\else)|(\\fi))  }
  \cs_set:Npn \colorizeifs {
   \regex_replace_all:NnN
     {\c_ifs_regex}
     {
       \c{codelinefont}[purple500]\cB\{\1\cE\}
     }\l__phdd_store_tl
}
  % Coffin l33 module. box vbox hbox use same colors
% captures hbox:n hbox_... etc
\clist_gset:Nn\coffinkeywords_clist
  { {(\\(coffin|box|hbox|vbox)_*[a-zA-Z_]+[^:])},
  }
\def\coffinkeywordsregex#1{
   \regex_replace_all:nnN
     {#1}
     {
       \c{codelinefont}[oprah]\cB\{\1\cE\}
     }\l__phdd_store_tl
}

% Key handlers such as |.code| common with \pkg{pgf}.
% reserved words starting with a dot and any number of letters terminating
% with a colon which is not colorized .code
% For pgf keys we can look for the equal sign to terminate the capture.
\regex_const:Nn \c_handlers_regex { ( ( [.]code|\.value|\.bool|.meta|.choice|\.tl )[a-
z\_]*[\:] )  }
  \cs_set:Npn \colorizehandlers {
   \regex_replace_all:NnN
     {\c_handlers_regex}
     {
       \c{codelinefont}[thered]\cB\{\1\cE\}
     }\l__phdd_store_tl
}
```

lorizemodulekeywords

To colorize modules such as **tex_**, **tl_** etc, we use a factory function. These are called via a clist, which stores the regexes. Best later to choose the colors from a color stack. #1 is the prefix and #2 is the style to be used. The style is to be changed to use a key value interface down the line. This can also be captured in a generic fashion.

```
\cs_generate_variant:Nn \clist_map_function:NN {cc}
\cs_generate_variant:Nn \regex_replace_all:NnN {cnN}
```

A list to hold simple regular expressions.

__phdd_regexes_clist

```
\clist_new:N \l__phdd_regexes_clist
```

Enables the addition of more keywords, so we can expose it at the programmer interface. The **\u** in the regular expression namely, **\u{htl var namei}** matches the exact contents of the token list htl vari. Within a **\c{...}** control sequence matching, the **\u** escape sequence only expands its argument once, in effect performing **\tl_to_str:v**. Quantifiers are not supported directly: use a group.

```
\cs_new_protected:Npn \regex_const:Nn #1#2
 {
   \__regex_compile:n {#2}
   \tl_const:Nx #1 { \exp_not:o \l__regex_internal_regex }
 }
```

The constant regex in l3regex is defined using `tl_const:Nx`, hence we build a new one from its definition to allow us to hook into the keywords list. We first handle modules. Our life is easier as all modules in l3 have prefixes with an underscore and can be captured fairly easily.

Token lists holding regex prefixes.

phd_more_keywords_tl

```
\cs_set:Npn \phd_more_keywords_tl {|phd}

\def\AAA#1{
 %  \regex_const:Nn \c_cs_regex
   \__regex_compile:n{(\\(cs|regex|seq|tl|clist|dim|char|int|%
group|bool|prg|iow|prop|exp|msg|quark|str|token|keys|q|c|%
coffin|box|hbox|vbox|tex|sys#1)_[a-zA-Z_]+[^:])}
  \tl_set:Nx \c_cs_regex { \exp_not:o \l__regex_internal_regex }
}
```

adds more module prefixes to `\c_cs_regex`

```
\cs_new:Npn \add_module_prefix:n #1 {
  \tl_put_right:Nn\phd_more_keywords_tl{#1}
  \expandafter\AAA\expandafter{\phd_more_keywords_tl }

}
\cs_generate_variant:Nn \add_module_prefix:n {x,f}
\NewDocumentCommand\AddPrefix{m}
  {
    \add_module_prefix:n {#1}
  }

\add_module_prefix:x {|phdd}


\tl_new:N \g__phdd_design_colors_tl
\tl_put_right:Nn \g__phdd_design_colors_tl
  {
    \cs_set:Npn \cscolor{\color{blue}}
    \cs_set:Npn \regexcolor{\color{theblue}}
    \cs_set:Npn \seqcolor{\color{purple500}}
    \cs_set:Npn \tlcolor{\color{blue900}}
    \cs_set:Npn \clistcolor{\color{purple800}}
    \cs_set:Npn \dimcolor{\color{unorange}}
    \cs_set:Npn \charcolor{\color{red}}
    \cs_set:Npn \intcolor{\color{red}}
    \cs_set:Npn \groupcolor{\color{red800}}
    \cs_set:Npn \boolcolor{\color{purple800}}
    \cs_set:Npn \prgcolor{\color{purple800}}
    \cs_set:Npn \iowcolor{\color{purple500}}
    \cs_set:Npn \propcolor{\color{red400}}
    \cs_set:Npn \expcolor{\color{redA100}}
```

```
    \cs_set:Npn \msgcolor{\color{green900}}
    \cs_set:Npn \quarkcolor{\color{red800}}
    \cs_set:Npn \strcolor{\color{orange700}}
    \cs_set:Npn \tokencolor{\color{orange700}}
    \cs_set:Npn \qcolor{\color{orange500}}
    \cs_set:Npn \ccolor{\color{unorange}}
    \cs_set:Npn \keyscolor{\color{deeporange700}}
    \cs_set:Npn \coffincolor{\color{deeporange700}}
    \cs_set:Npn \boxcolor{\color{deeporange600}}
    \cs_set:Npn \hboxcolor{\color{deeporange500}}
    \cs_set:Npn \vboxcolor{\color{deeporange500}}
    \cs_set:Npn \phdcolor{\color{deeporange900}}
    \cs_set:Npn \phddcolor{\color{deeporange900}}
    \cs_set:Npn \texcolor{\color{red800}}
    \cs_set:Npn \cctabcolor{\color{red800}}
    \cs_set:Npn \intarraycolor{\color{green500}}
    \cs_set:Npn \syscolor{\color{purple500}}
  }

\tl_use:N \g__phdd_design_colors_tl

\regex_const:Nn \c_tl_regex {((\\tl_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+tl)) }
\regex_const:Nn \c_seq_regex {((\\seq_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+seq)) }
\regex_const:Nn \c_clist_regex { ((\\clist_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+clist)) }
\regex_const:Nn \c_tex_regex { (\\tex_[a-zA-Z_]+[^:]) }
\regex_const:Nn \c_exp_regex { (\\exp_[a-zA-Z_]+[^:]) }
\regex_const:Nn \c_dim_regex { ((\\bool_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+dim)) }
\regex_const:Nn \c_str_regex { ((\\str_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+str)) }
\regex_const:Nn \c_bool_regex { ((\\bool_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+bool)) }
\regex_const:Nn \c_char_regex { ((\\char_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+char)) }
\regex_const:Nn \c_int_regex { ((\\int_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+int)) }
\regex_const:Nn \c_iow_regex { (\\iow_[a-zA-Z_]+[^:]) }
\regex_const:Nn \c_group_regex { (\\group_[a-zA-Z_]+[^:]) }
\regex_const:Nn \c_box_regex { (\\box_[a-zA-Z_]+[^:]) }
\regex_const:Nn \c_skip_regex { ((\\skip_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+skip)) }
\regex_const:Nn \c_token_regex { ((\\token_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+token)) }
\regex_const:Nn \c_regex_regex { ((\\regex_[a-zA-Z_]+[^:])|(\\c_[^_][a-z\_]+regex)) }

\cs_new:Npn \colorizemodulekeywords #1 #2
  {
    \cs_set:cpn {currentcolors} {\color{#2}}

    \cs_gset:cpn {#1keywordsregex} ##1
```

```
      {
     \regex_replace_all:nnN
      { ##1}
      {
        \c{codelinefont}[#2]\cB\{\0\cE\}
      }\l__phdd_store_tl
      }

      \cs_gset:Npn \keywordsregex_aux:n ##1 ##2
      {
     \regex_replace_all:cnN
        {c_cs_regex}
      {
        \cB{
           \c{\2color}\1
        \cE}
      }\l__phdd_store_tl
      }
```

In order to speed up the regular expression processing we predefine common module regular expressions

```
    \cs_if_free:cTF {c_cs_regex} {
%      \clist_gset:cn {#1keywords_clist}
%        { % c if it has a constant c_underscore_str, i.e it has no :
%          {(\\(#1)_[a-zA-Z_]+[^:])},
%          % capture c_something_tl but not c__something_tl
%          {\\c_[^_][a-z\_]+#1}
%        }
%      \clist_map_function:cc {#1keywords_clist} {#1keywordsregex}
    }
    { \keywordsregex_aux:n {#1} {#2}  }

  }
```

(*End definition for* `\colorizemodulekeywords`, `\l__phdd_regexes_clist`, *and* `\phd_more_keywords_tl`.)

`keywordsregex` {⟨*regex expression for keywords*⟩}

```
%\def\keywordsregex#1{
%    \regex_replace_all:nnN
%      {#1}
%      {
%        \c{codelinefont}[blue]\cB\{\0\cE\}
%      }\l__phdd_store_tl
%}
```

(*End definition for* `keywordsregex`.)

`\colorizebrackets`
`\colorizesquarebrackets`

```
\regex_const:Nn \c_braces_regex { (\{|\})+ }
\regex_const:Nn \c_square_brackets_regex { [\[\]]+ }
\regex_const:Nn \c_brackets_regex { [\(\)] }

\def\bracescolor#1{\normalcolor\textcolor{thebracescolor}{#1}\normalcolor}

```

```
\def\squarebracketscolor#1{\textcolor{thesquarebracketscolor}{#1}}

\def\bracketscolor#1{\bgroup\bfseries\color{thered}#1\egroup}
\cs_set:Npn \colorizebraces #1
  {
  \regex_replace_all:NnN
    {
      \c_braces_regex
    }
    {
      \c{bracescolor}\cB\{\0\cE\}
    }\l__phdd_store_tl
  }

\cs_set:Npn \colorizesquarebrackets #1
  {
  \regex_replace_all:NnN
    {
      \c_square_brackets_regex
    }
    {
      \c{squarebracketscolor}\cB\{\0\cE\}
    }\l__phdd_store_tl
  }
\cs_set:Npn \colorizebrackets #1
  {
  \regex_replace_all:NnN
    {
      \c_brackets_regex
    }
    {
      \c{bracketscolor}\0
    }\l__phdd_store_tl
  }

% Highlight math operators
% First we do the math operator signs$
\regex_const:Nn \c_math_operator_regex { [\$]+ }

\def\mathoperatorcolor#1
{
  \bgroup
    \bfseries\color{thered}#1
  \egroup
}

\cs_set:Npn \colorizemathoperator #1
  {
  \regex_replace_all:NnN
    {
      \c_math_operator_regex
    }
    {
      \c{mathoperatorcolor}\cB\{\0 \cE\}
```

```
    }\l__phdd_store_tl
  }
```

(*End definition for* `\colorizebrackets` *and* `colorizesquarebrackets`.)

```
\cs_new:Npn \colorizexparsedeclarations{
 \regex_replace_all:nnN
     {
       \\(DeclareDocumentCommand|DeclareOption|NewDocumentCommand%
         |DeclareRobustCommand|NewDocumentEnvironment|r*e*newcommand|newenvironment|def)
     }
     {
        \c{codelinefont}[blue]\cB\{\1\cE\}
     }\l__phdd_store_tl
}
```

The command colorizes the output in macro code. It uses the l3regex package, which is inherently slow. It is only called when the key `/phdd/codecolorize` is set to true.

`d_code_colorize_bool`

```
%\bool_new:N \phdd_code_colorize_bool
%\bool_gset_false:N \phdd_code_colorize_bool
```

(*End definition for* `\phdd_code_colorize_bool`.)

`colorizecomments`

```
% Attempt to capture comment strings
% for TeX these are the (%). Note the (\v) in the
% regex is any character for vertical space ^^M.
% Also captures \foo_underscore:N.
% Captures also escape characters # & * - and others.
% Parameter parameter_test
%%
% % %
% Capture all characters up to vertical command. Has issues
% as it is not protected and other commands can colorize it.
%
%
% \ \*  Any number of starting spaces
%
% [^\<]* Do not colorize guards i.e., %<*guard>, use star as
% we may have only a %.
% Strategy: capture anything except end of lines. These are
% modified by \obeylines and \doc uses everypar that makes
% it hard to capture.
\regex_const:Nn \c_comments_regex
  {
    (\%\ *\%*[ a-zA-Z\ \\\*\_\:\;\.\!\"\`\'\#\&\-
\^\(\)\|\d\=\+\,\@\}\{\[\]\/\% ]*)
  }

\cs_gset:Npn \colorizecomments
  {
      \regex_replace_all:NnN \c_comments_regex
     {
```

```
          \c{color}\cB{green800\cE}\1\c{normalcolor}
        }\l__phdd_store_tl
  }

```

(*End definition for* `colorizecomments`.)

```

\cs_gset:Npn \listings_code
  {
  %\colorizebraces{orangeA700}
%  \colorizesquarebrackets{orange500}
  %\colorizebrackets{thered}
%  \colorizemathoperator{thered}
%%
%   \colorizehandlers
%  \colorizemodulekeywords{tex}{red500}
%  \colorizemodulekeywords{regex}{red400}
%  \colorizemodulekeywords{exp}{red500}
%  \colorizemodulekeywords{dim}{teal500}
%  \colorizemodulekeywords{mode}{teal500}
%  \colorizemodulekeywords{clist}{teal600}
%  \colorizemodulekeywords{str}{unorange}
%  \colorizemodulekeywords{bool}{blue800}
%  \colorizemodulekeywords{tl}{blue600}
%
%  \colorizemodulekeywords{token}{orange700}
%
  \colorizemodulekeywords{cs}{-thecsmodulecolor}
%  \colorizemodulekeywords{xcol}{thecsmodulecolor}
%  \colorizemodulekeywords{char}{orange900}
%  \colorizemodulekeywords{seq}{blue900}
%  \colorizemodulekeywords{iow}{blueprint}
%  \colorizemodulekeywords{int}{purple500}
%  \colorizemodulekeywords{dim}{purple500}
%  \colorizemodulekeywords{tex}{unorange}
%  \colorizemodulekeywords{skip}{purple700}
%  \colorizemodulekeywords{hbox}{purple500}
%  \colorizemodulekeywords{box}{purple500}
%  \colorizemodulekeywords{group}{purple500}
%  \colorizeifs{}
  %\colorizecomments
  %\commentshack
 % \show\l__phdd_store_tl

  }
\ExplSyntaxOff
/colorize
*DOCUM

\ExplSyntaxOn
\group_begin:
  \char_set_catcode_other:N \^^A
  \char_set_catcode_active:N \^^S
  \char_set_catcode_active:N \^^B
```

```
  \char_set_catcode_other:N \^^L
  \char_set_catcode_other:N \^^R
  \char_set_lccode:nn { `\^^A } { `\% }
  \char_set_lccode:nn { `\^^S } { `\  }
  \char_set_lccode:nn { `\^^B } { `\\ }
  \char_set_lccode:nn { `\^^L } { `\{ }
  \char_set_lccode:nn { `\^^R } { `\} }
  \tex_lowercase:D
    {
      \group_end:
      \cs_set_protected:Npn \xmacro@code #1 ^^A ^^S^^S^^S^^S ^^Bend ^^Lmacrocode^^R
        {\__phdd_xmacro_code:n {#1} \end{macrocode} }
    }


\group_begin:
  \bool_set_false:N \phdd_code_colorize_bool
  \char_set_catcode_active:N \<
  \char_set_catcode_active:N \>
```

We check for guards as well as <nocolor> directive

```
  \cs_new_protected:Npn \__phdd_xmacro_code:n #1
    {
      \tl_if_in:nnTF {#1} { < @ @ = }
        { \__phdd_xmacro_code:w #1 \q_stop }
        {
          \tl_if_in:nnTF {#1} { < nocolor }
          {
            \bool_set_false:N \phdd_code_colorize_bool
            \tl_gset:Nn \l__phdd_store_tl {#1}
            \__phdd_detect_internals:N \l__phdd_store_tl
            \__phdd_replace_at_at:N \l__phdd_store_tl
            % Call the pretty printer or skip

            \bool_if:NTF \phdd_code_colorize_bool
              {
                \listings_code
                \tl_use:N \l__phdd_store_tl
              }
            {#1}

          }
          {
            \tl_gset:Nn \l__phdd_store_tl {#1}
            \__phdd_detect_internals:N \l__phdd_store_tl
            \__phdd_replace_at_at:N \l__phdd_store_tl
            % Call the pretty printer or skip

            \bool_if:NTF \phdd_code_colorize_bool
              {
                \listings_code
                \tl_use:N \l__phdd_store_tl
              }
            {
```

```
              \tl_use:N \l__phdd_store_tl


            % #1
             }
           }
         }

%        \tl_if_in:nnTF {#1} { < nocolor }
%        { \__phdd_xmacro_code:w #1 \q_stop }
%        {
%          \tl_gset:Nn \l__phdd_store_tl {#1}
%          \__phdd_detect_internals:N \l__phdd_store_tl
%          \__phdd_replace_at_at:N \l__phdd_store_tl
%          % Call the pretty printer or skip
%          \bool_set_true:N \phdd_code_colorize_bool
%          \bool_if:NTF \phdd_code_colorize_bool
%            {
%              \listings_code
%              \tl_use:N \l__phdd_store_tl
%            }
%            {#1}
%        }

    }




  \cs_new_protected:Npn \__phdd_xmacro_code:w #1 < @ @ = #2 > #3 \q_stop
    {
      \tl_set:Nn \l__phdd_tmpa_tl {#1}
      \__phdd_detect_internals:N \l__phdd_tmpa_tl
      \__phdd_replace_at_at:N \l__phdd_tmpa_tl

% We have found the name hence store it
      \tl_gset:Nn \g__phdd_module_name_tl {#2}
      \tl_put_right:Nn \l__phdd_tmpa_tl { < @ @ = #2 > }

% three reprsents the full code outside the guards
      \tl_set:Nn \l__phdd_tmpb_tl {#3}
% detect interna;
      \__phdd_detect_internals:N \l__phdd_tmpb_tl
% replace ats
      \__phdd_replace_at_at:N \l__phdd_tmpb_tl
%
      \tl_put_right:No \l__phdd_tmpa_tl { \l__phdd_tmpb_tl }
      \tl_use:N \l__phdd_tmpa_tl
    }


\group_end:
```

D.2.6 At end document

Print all defined and documented macros/functions.

```
\iow_new:N \g__phdd_func_iow
\tl_new:N \l__phdd_doc_def_tl
\tl_new:N \l__phdd_doc_undef_tl
\tl_new:N \l__phdd_undoc_def_tl
\cs_new_protected:Npn \__phdd_show_functions_defined:
  {
    \bool_lazy_and:nnT
      { \g__phdd_typeset_implementation_bool } { \g__phdd_checkfunc_bool }
      {
        \iow_term:x { \c__phdd_iow_separator_tl \iow_newline: }
        \iow_open:Nn \g__phdd_func_iow { \c_sys_jobname_str .cmds }

        \tl_clear:N \l__phdd_doc_def_tl
        \tl_clear:N \l__phdd_doc_undef_tl
        \tl_clear:N \l__phdd_undoc_def_tl
        \seq_map_inline:Nn \g_doc_functions_seq
          {
            \seq_if_in:NnTF \g_doc_macros_seq {##1}
              {
                \tl_put_right:Nx \l__phdd_doc_def_tl
                  { ##1 \iow_newline: }
                \iow_now:Nn \g__phdd_func_iow { > ~ ##1 }
              }
              {
                \tl_put_right:Nx \l__phdd_doc_undef_tl
                  { ##1 \iow_newline: }
                \iow_now:Nn \g__phdd_func_iow { ! ~ ##1 }
              }
          }
        \seq_map_inline:Nn \g_doc_macros_seq
          {
            \seq_if_in:NnF \g_doc_functions_seq {##1}
              {
                \tl_put_right:Nx \l__phdd_undoc_def_tl
                  { ##1 \iow_newline: }
                \iow_now:Nn \g__phdd_func_iow { ? ~ ##1 }
              }
          }
        \__phdd_functions_typeout:nN
          {
            Functions~both~documented~and~defined: \iow_newline:
            (In~order~of~being~documented)
          }
          \l__phdd_doc_def_tl
        \__phdd_functions_typeout:nN
          { Functions~documented~but~not~defined: }
          \l__phdd_doc_undef_tl
        \__phdd_functions_typeout:nN
          { Functions~defined~but~not~documented: }
          \l__phdd_undoc_def_tl
```

```

        \iow_close:N \g__phdd_func_iow
        \iow_term:x { \c__phdd_iow_separator_tl }
      }
  }
\AtEndDocument { \__phdd_show_functions_defined: }
```

TODO: use `\iow_term:x`.

```
\cs_new_protected:Npn \__phdd_functions_typeout:nN #1#2
  {
    \tl_if_empty:NF #2
      {
        \typeout
          {
            \c__phdd_iow_midrule_tl \iow_newline:
            #1 \iow_newline:
            \c__phdd_iow_midrule_tl \iow_newline:
            #2
          }
        \tl_clear:N #2
      }
  }
```

```
\cs_new_protected:Npn \__phdd_show_not_tested:
  {
    \bool_if:NT \g__phdd_checktest_bool
      {
        \tl_clear:N \l__phdd_tmpa_tl
        \prop_if_empty:NF \g__phdd_missing_tests_prop
          {
            \cs_set:Npn \__phdd_tmpa:w ##1##2
              {
                \iow_newline:
                \space\space\space\space \exp_not:n {##1}
                \clist_map_function:nN {##2} \__phdd_tmpb:w
              }
            \cs_set:Npn \__phdd_tmpb:w ##1
              {
                \iow_newline:
                \space\space\space\space\space\space * ~ ##1
              }
            \tl_put_right:Nx \l__phdd_tmpa_tl
              {
                \iow_newline: \iow_newline:
                The~ following~ macro(s)~ have~ incomplete~ tests:
                \iow_newline:
                \prop_map_function:NN
                  \g__phdd_missing_tests_prop \__phdd_tmpa:w
              }
          }
        \seq_if_empty:NF \g__phdd_not_tested_seq
          {
            \cs_set:Npn \__phdd_tmpa:w ##1
              { \clist_map_function:nN {##1} \__phdd_tmpb:w }
            \cs_set:Npn \__phdd_tmpb:w ##1
```

```
                {
                  \iow_newline:
                  \space\space\space\space ##1
                }
              \tl_put_right:Nx \l__phdd_tmpa_tl
                {
                  \iow_newline:
                  \iow_newline:
                  The~ following~ macro(s)~ do~ not~ have~ any~ tests:
                  \iow_newline:
                  \seq_map_function:NN
                    \g__phdd_not_tested_seq \__phdd_tmpa:w
                }
            }
          \tl_if_empty:NF \l__phdd_tmpa_tl
            {
              \int_set:Nn \l__phdd_tmpa_int { \tex_interactionmode:D }
              \errorstopmode
              \ClassError { l3doc } { \l__phdd_tmpa_tl } { }
              \int_set:Nn \tex_interactionmode:D { \l__phdd_tmpa_int }
            }
        }
    }
\AtEndDocument { \__phdd_show_not_tested: }
```

D.2.7 Indexing

D.2.70 Userspace commands

Fix index (for now):

```
\g@addto@macro \theindex { \MakePrivateLetters }
\cs_gset:Npn \verbatimchar {&}
\setcounter { IndexColumns } { 2 }
```

Set up the Index to use `\part`

```
\IndexPrologue
  {
    \part*{Index}
    \markboth{Index}{Index}
    \addcontentsline{toc}{part}{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.
  }
```

`\SpecialIndex` An attempt at affecting how commands which appear within the `macrocode` environment are treated in the index.

```
\cs_gset_protected:Npn \SpecialIndex #1
  {
    \@bsphack
    \__phdd_special_index:nn {#1} { }
```

```
    \@esphack
  }
```

(*End definition for* `\SpecialIndex`.)

```
\msg_new:nnn { l3doc } { print-index-howto }
  {
    Generate~the~index~by~executing\\
    \iow_indent:n
      { makeindex~-s~gind.ist~-o~\c_sys_jobname_str.ind~\c_sys_jobname_str.idx }
  }
\tl_gput_right:Nn \PrintIndex
  { \AtEndDocument { \msg_info:nn { l3doc } { print-index-howto } } }
```

D.2.70 Internal index commands

`\it@is@a` The index of one-character commands within the `macrocode` environment is produced using `\it@is@a` ⟨*char*⟩. Alter that command.

```
\cs_gset_protected:Npn \it@is@a #1
  {
    \use:x
      {
        \__phdd_special_index_module:nnnnN
          {#1}
          { \bslash #1 }
          { }
          { }
          \c_false_bool
      }
  }
```

(*End definition for* `\it@is@a`.)

`\__phdd_special_index:nn`

```
\cs_new_protected:Npn \__phdd_special_index:nn #1#2
  {
    \__phdd_key_get:n {#1}
    \quark_if_no_value:NF \l__phdd_override_module_tl
      { \tl_set_eq:NN \l__phdd_index_module_tl \l__phdd_override_module_tl }
    \__phdd_special_index_module:ooonN
      { \l__phdd_index_key_tl }
      { \l__phdd_index_macro_tl }
      { \l__phdd_index_module_tl }
      {#2}
      \l__phdd_index_internal_bool
  }
\cs_generate_variant:Nn \__phdd_special_index:nn { o }
```

(*End definition for* `\__phdd_special_index:nn`.)

`l_index_module:nnnnN`
`l_index_module:ooonN`
`ial_index_aux:nnnnnn`
`special_index_set:Nn`

Remotely based on Heiko's replacement to play nicely with **hypdoc**. We use `\verb` or a `\verbatim@font` construction depending on whether the number of tokens in `#2` is equal to its number of characters: if it is not then that suggests that there is a construct such as `\meta{...}`.

```
\tl_new:N \l__phdd_index_escaped_macro_tl
\tl_new:N \l__phdd_index_escaped_key_tl
\cs_new_protected:Npn \__phdd_special_index_module:nnnnN #1#2#3#4#5
```

#1 = key
#2 = macro
#3 = module
#4 = index 'type' (`main`/`usage`/*etc.*)
#5 = boolean whether internal command

```
  {
    \use:x
      {
        \exp_not:n { \__phdd_special_index_aux:nnnnnn {#1} {#2} }
          \tl_if_empty:nTF {#3}
            { { } { } { } }
            {
              \str_if_eq:nnTF {#3} { TeX }
                {
                  { TeX~and~LaTeX2e }
                  { \string\TeX{}~and~\string\LaTeXe{} }
                }
                {
                  {#3}
                  { %\string\pkg our \pkg is a ref use something else
                   \bgroup\string\color{red}
                   \string\texttt{#3}~\egroup }
                }
              { \bool_if:NT #5 { ~internal } ~commands: }
            }
      }
          {#4}
  }
\cs_generate_variant:Nn \__phdd_special_index_module:nnnnN { ooo }
\cs_new_protected:Npn \__phdd_special_index_aux:nnnnnn #1#2#3#4#5#6
```

#1 = key
#2 = macro
#3 = index subheading string
#4 = index subheading text
#5 = index subheading suffix (appended to both arg 3 and 4)
#6 = index 'type' (`main`/`usage`/*etc.*)

```
  {
    \tl_set:Nn \l__phdd_index_escaped_key_tl {#1}
    \__phdd_quote_special_char:N \l__phdd_index_escaped_key_tl
    \__phdd_special_index_set:Nn \l__phdd_index_escaped_macro_tl {#2}
    \str_if_eq:onTF { \@currenvir } { macrocode }
      { \codeline@wrindex }
      {
        \str_case:nnF {#6}
          {
            { main }  { \codeline@wrindex }
            { usage } { \index }
```

```
          }
          { \HD@target \index }
      }
      {
        \tl_if_empty:nF { #3 #4 #5 }
          { #3 #5 \actualchar #4 #5 \levelchar }
        \l__phdd_index_escaped_key_tl
        \actualchar
        {
          \token_to_str:N \verbatim@font \c_space_tl
          \l__phdd_index_escaped_macro_tl
        }
        \encapchar
        hdclindex{\the\c@HD@hypercount}{#6}
      }
  }
\cs_new_protected:Npn \__phdd_special_index_set:Nn #1#2
  {
    \tl_set:Nx #1 { \tl_to_str:n {#2} }
    \__phdd_if_almost_str:nTF {#2}
      {
        \tl_replace_all:Non #1 { \tl_to_str:n { __ } }
          {
            \verbatimchar
            \token_to_str:N \_ \token_to_str:N \_
            \token_to_str:N \verb * \verbatimchar
          }
        \exp_args:Nx \tl_map_inline:nn
          { \tl_to_str:N \verbatimchar \token_to_str:N _ }
          {
            \tl_replace_all:Nnn #1 {##1}
              {
                \verbatimchar \c_backslash_str ##1
                \token_to_str:N \verb * \verbatimchar
              }
          }
        \tl_set:Nx #1
          {
            \token_to_str:N \verb * \verbatimchar
            #1 \verbatimchar
          }
      }
      {
        \tl_set:Nn #1 {#2}
        \tl_replace_all:Non #1
          { \c_backslash_str }
          { \token_to_str:N \bslash \c_space_tl }
      }
    \__phdd_quote_special_char:N #1
  }
```

(*End definition for* `\__phdd_special_index_module:nnnnN`, `\__phdd_special_index_aux:nnnnnn`, *and* `\__phdd_special_index_set:Nn`.)

quote_special_char:N

Quote some special characters.

```
\cs_new_protected:Npn \__phdd_quote_special_char:N #1
  {
    \tl_map_inline:nn { \quotechar \actualchar \encapchar \levelchar }
      {
        \tl_replace_all:Nxn #1
          { \tl_to_str:N ##1 } { \quotechar \tl_to_str:N ##1 }
      }
  }
```

(*End definition for* `\__phdd_quote_special_char:N`.)

D.2.70 Finding sort-key and module

__phdd_key_get:n

Sets **\l__phdd_index_macro_tl**, **\l__phdd_index_key_tl**, and **\l__phdd_index_module_tl** from **#1**. The base function is stored by **__phdd_key_get_base:nN** in **\l__phdd_index_macro_tl**, falling back to **#1** if it contains markup or has no signature.

The starting point for the ⟨*key*⟩ is **\l__phdd_index_key_tl** as a string. If it the first character is a backslash, remove it. Then recognize **expl** functions and variables by the presence of **:** or **_** and TEX/LATEX 2_ε commands by the presence of **@**. For **expl** names, we call **__phdd_key_func:** or **__phdd_key_var:**, which are responsible for removing some characters and finding the module name, while for TEX/LATEX 2_ε commands the module name is **TeX**, and others have an empty module name.

```
\cs_new_protected:Npn \__phdd_key_get:n #1
  {
    \__phdd_key_get_base:nN {#1} \l__phdd_index_macro_tl
    \tl_set:Nx \l__phdd_index_key_tl
      { \tl_to_str:N \l__phdd_index_macro_tl }
    \tl_clear:N \l__phdd_index_module_tl
    \tl_if_in:NoTF \l__phdd_index_key_tl { \tl_to_str:n { __ } }
      { \bool_set_true:N \l__phdd_index_internal_bool }
      { \bool_set_false:N \l__phdd_index_internal_bool }
    \exp_last_unbraced:NNo
    \tl_if_head_eq_charcode:oNT
      { \l__phdd_index_key_tl } \c_backslash_str
      { \__phdd_key_pop: }
    \tl_if_in:NoTF \l__phdd_index_key_tl { \token_to_str:N : }
      { \__phdd_key_func: }
      {
        \tl_if_in:NoTF \l__phdd_index_key_tl { \token_to_str:N _ }
          { \__phdd_key_var: }
          {
            \tl_if_in:NoT \l__phdd_index_key_tl { \token_to_str:N @ }
              { \tl_set:Nn \l__phdd_index_module_tl { TeX } }
          }
      }
  }
\cs_new_protected:Npn \__phdd_key_pop:
  {
    \tl_set:Nx \l__phdd_index_key_tl
```

```
      { \tl_tail:N \l__phdd_index_key_tl }
  }
```

(*End definition for* `\__phdd_key_get:n`.)

`dd_key_trim_module:n`
`ey_drop_underscores:`

Helper that removes from **\l__phdd_index_module_tl** everything after the first occurence of **#1**. Helper that removes any leading underscore from **\l__phdd_index_key_tl**.

```
\cs_new_protected:Npn \__phdd_key_trim_module:n #1
  {
    \cs_set:Npn \__phdd_tmpa:w ##1 #1 ##2 \q_stop
      { \exp_not:n {##1} }
    \tl_set:Nx \l__phdd_index_module_tl
      { \exp_after:wN \__phdd_tmpa:w \l__phdd_index_module_tl #1 \q_stop }
  }
\cs_new_protected:Npn \__phdd_key_drop_underscores:
  {
    \tl_if_head_eq_charcode:oNT { \l__phdd_index_key_tl } _
      { \__phdd_key_pop: \__phdd_key_drop_underscores: }
  }
```

(*End definition for* `\__phdd_key_trim_module:n` *and* `\__phdd_key_drop_underscores:`.)

`\__phdd_key_func:`

The function **__phdd_key_func:** is used if there is a colon, so either for usual **expl3** functions or for keys from **l3keys**. After removing from the key a leading dot (for the latter case), and any leading underscore, the module name is the part before any colon or underscore.

```
\cs_new_protected:Npn \__phdd_key_func:
  {
    \tl_if_head_eq_charcode:oNT { \l__phdd_index_key_tl } .
      { \__phdd_key_pop: }
    \__phdd_key_drop_underscores:
    \tl_set_eq:NN \l__phdd_index_module_tl \l__phdd_index_key_tl
    \exp_args:No \__phdd_key_trim_module:n { \token_to_str:N : }
    \exp_args:No \__phdd_key_trim_module:n { \token_to_str:N _ }
  }
```

(*End definition for* `\__phdd_key_func:`.)

`\__phdd_key_var:`
`phdd_key_get_module:`

The function **__phdd_key_var:** covers cases with no **:** but with **_**, typically variables but occasionally non-**expl3** functions such as Lua function with underscores. First test the second character: if that is **_** then assume we have a proper variable, otherwise use the part before any underscore as the module name. For variables, distinguish quarks and scan marks (starting with **q** and **s**), then drop the first letter (local/global/constant marker) and underscores. If there is no underscore left we had something like **\c_zero** which we assume is an integer constant. If there is one underscore we assume it is a variable like **\c_empty_tl** whose module name is the last part. Otherwise the module name is the part before any underscore.

```
\cs_new_protected:Npn \__phdd_key_var:
  {
    \exp_args:Nx \tl_if_head_eq_charcode:nNTF
      { \exp_args:No \str_tail:n \l__phdd_index_key_tl } _
      {
```

```
        \str_case:fn { \str_head:N \l__phdd_index_key_tl }
          {
            { q } { \tl_set:Nn \l__phdd_index_module_tl { quark } }
            { s } { \tl_set:Nn \l__phdd_index_module_tl { quark } }
          }
        \__phdd_key_pop:
        \__phdd_key_pop:
        \__phdd_key_drop_underscores:
        \tl_if_empty:NT \l__phdd_index_module_tl
          {
            \seq_set_split:NoV \l__phdd_tmpa_seq
              { \token_to_str:N _ } \l__phdd_index_key_tl
            \tl_set:Nx \l__phdd_index_module_tl
              {
                \int_case:nnF { \seq_count:N \l__phdd_tmpa_seq }
                  {
                    { 0 } { }
                    { 1 } { int }
                    { 2 } { \seq_item:Nn \l__phdd_tmpa_seq { 2 } }
                  }
                  { \seq_item:Nn \l__phdd_tmpa_seq { 1 } }
              }
          }
      }
      {
        \tl_set_eq:NN \l__phdd_index_module_tl \l__phdd_index_key_tl
        \exp_args:No \__phdd_key_trim_module:n { \token_to_str:N _ }
      }
  }
```

(*End definition for* `\__phdd_key_var:` *and* `\__phdd_key_get_module:`.)

D.2.8 Change history

Set the change history to use `\part`. Allow control names to be hyphenated in here...

```
\GlossaryPrologue
  {
    \part*{Change~History}
    {\GlossaryParms\ttfamily\hyphenchar\font=`\-}
    \markboth{Change~History}{Change~History}
    \addcontentsline{toc}{part}{Change~History}
  }
```

```
\msg_new:nnn { phdd } { print-changes-howto }
  {
    Generate~the~change~list~by~executing\\
    \iow_indent:n
      { makeindex~-s~gglo.ist~-o~\c_sys_jobname_str.gls~
                                  \c_sys_jobname_str.glo }
  }
\tl_gput_right:Nn \PrintChanges
  { \AtEndDocument { \msg_info:nn { phdd } { print-changes-
howto } } }
```

```
%
```

Allocate a counter and use LaTeX `\list` to implement this environment. For example:

#1 = first argument
#2 = second argument
#3 = third argument Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

`labelargcount`

```
\newcounter{phddargcount}
\cs_new:Npn \labelargcount
  {
     \texttt{\#\arabic{phddargcount}\hskip\labelsep$=$}
  }
```

(*End definition for* `labelargcount`.)

`\macroarguments`
`\endmacroarguments`

```
\cs_new:Npn \macroarguments
  {\list\labelargcount
    {\usecounter{phddargcount}
     \leftmargin=2\leftmargin
     \parsep \z@ \@plus\z@ \@minus\z@
     \topsep 4\p@ \@plus\p@ \@minus2\p@
     \itemsep\z@ \@plus\z@ \@minus\z@
     \def\makelabel##1{\hss\llap{##1}}}
  }

\cs_new:Npn \endmacroaruments {\endlist}
```

(*End definition for* `\macroarguments` *and* `\endmacroarguments`.)

`\AltMacroFont` Set doc's font for modules to up, otherwise * cannot be printed in unicode fonts (we have issues...) [1] TODO change to lmtt

```
\def\Module{
 \color{green800}
}
\def\AltMacroFont{
 \fontfamily{lmtt}\fontseries{lc}
 }
```

(*End definition for* `\AltMacroFont`.)

```
\wlog{*************************************************}
\wlog{ END PHD-DOCUMENTATION                            }
\wlog{*************************************************}

/DOCUM
```

[2]

[1] 1.01 2018/01/18 Added settings.
[2] 1.0,2013/01/26,Converted to DTX file.

Code Implementation and Objectives

The package provides macros for documenting and indexing primarily LaTeX or LaTeX 3 code. It combines the styles originitating with `doc` with those found in the `pgf` series of packages. It removes some of the difficulties of using hyperlinks in the index and elsewhere in the document. 21.00002

16

-2.5

-2.1

0.87865

Having provided for individual control for virtually all document elements, we need some form of organizing all the color keys. This package defines different color palettes, that make it easier to change all the colors of a document with a single key setting (provided of course we have a predefined palette).

We start by outlining what we are trying to achieve with the **phd-colorpalette** package.

1. To provide a declarative interface to enable users to modify colors specifications in logical groups rather than individual elements..
2. To provide a number of predefined color themes.
3. To provide a plug-in architecture for user extensions.

E.1 Terminology

document Any written item, as a book, article, or letter, especially of a factual or informative nature.

heading A division of a document or document series. For a normal book headings are chapters, sections etc. However we allow for specifying a more complex document divided into books, volumes parts etc. For example the Bible has Books, chapters and verses, where a legal document might require divisions such as clauses. In general these divisions are numbered. These document divisions are stored in the comma list

head A typeset heading, such as chapter head, or section head. This can include a counter, label and title for example, *Chapter 1 Introduction*.

dom This is a programming interface that provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed. Although LaTeX does not offer a standard way to build such a tree (mainly because TeX does not require the marking of paragraphs, it is useful to think of the document as a tree structure. We also allow for a semi-automated way to build such a tree (with the exception that paragraphs are not included).

element A part of the document tree that can be styled on its own. For example the chapter label, or the section number.

E.2 Users

We classify users according to the LaTeX3 terminology as a) programmers b) template designers and c) authors.

E.2.1 Author

We assume that the author has an exising template which she is using but might want to do some minor modifications, for example use an italic shape for the font of the mark, but an upright font for the page numbers.

We follow the idea of representing the basic elements of documents as elements, each one having a parent in order to specify the element we need to style as accurate as possible. One can think of this approach being congruent with objects in other languages. As a matter fact nothing stops us from defining a key value interface as shown below.

E.2.2 Template designer

The template designer in the example above would have selected the format style from a number of predefined formats (templates) or would have created a style called *apa* from an existing template and modified it using declarative key style.

E.2.3 The Programmer

The programmer in the example above could have created the basic format *apa* by using both declarative as well as defining or using existing macros. To the programmer we offer an extension mechanism, where the contents of a `ps@` command are defined. For example the programmer can define a new style using Ti*k*Z, but without having to worry about defining full `ps@` and their interface.

E.3 Color Palettes

Although documents would normally have only a small number of colors, the number of variables that need a color setting is quite large. In this package we introduce the option of a *color palette*, where all color settings can be in one place.

E.4 Companion Go packages

The Companion Go **colornames.go** package can be used to generate Material Design color palettes inspired by Google's material design. Color in Material Design uses bold hues juxtaposed with muted environments, deep shadows, and bright highlights.

Palettes provided with the package have memorable names.

```
\cxset{palette bbc}
```

Once a palette is set a number of colornames are available and are used by the rest of the phd packages to set colors.

`\cxset` {⟨*palette name*⟩} For example setting the palette to *unorange*[1] will change *thesubsectioncolor* to an orange color defined with the *unorange* palette.

This renders as follows:

[1] unbelievable orange

Example 4.1 Setting palette colors

```
% \bgroup
% \cxset{palette unorange}
% \subsection{Subsection heading example}
% \subsubsection{Subsubsection example}
% \lorem
% \egroup
%
```

E.4.1 Subsection heading example

E.4.10 Subsubsection example

Fusce adipiscing justo nec ante. Nullam in enim. Pellentesque felis orci, sagittis ac, malesuada et, facilisis in, ligula. Nunc non magna sit amet mi aliquam dictum. In mi. Curabitur sollicitudin justo sed quam et quadd.par

E.5 Preliminaries

Standard file identification. We first announce the package and require that it be used with LaTeX2e. need to upgrade

```
\NeedsTeXFormat{LaTeX2e}[2017/04/15]%
\ProvidesPackage{phd-colorpalette}[2018/1/13 v1.0 color palettes (YL)]%
\RequirePackage{phd-materialdesign}
```

Ensure we have the **xcolor**

```
\@ifpackageloaded{xcolor}{}%
 {\PassOptionsToPackage{\xcolorkeys@cx}{xcolor}
  \RequirePackage{xcolor}}
\ExplSyntaxOn
\cs_set_eq:NN \xcol_define_color:nnn \definecolor
\cs_set_eq:NN \xcol_let:nn \colorlet
\xcol_define_color:nnn { themacrocolor } { rgb } { 0.784, 0.06, 0.176 }
\ExplSyntaxOff
```

\docColorBox{⟨*color name*⟩}

Example: A demo box for the `\docColor{thegreen}` `\docColorBox{thegreen}`.
A demo box for the `thegreen`.

```
\long\def\docColorBox#1{%
\tcbox[on line,arc=0pt,outer arc=0pt,colback=#1,colframe=white,
   boxsep=0pt,left=0pt,right=0pt,top=1pt,bottom=1pt,
   boxrule=0pt,bottomrule=1pt,toprule=1pt]{\Zi}%
}
```

E.6 Palette and Color Naming Conventions

One of the conventions used in the **phd** series of packages is that mostly names are lower cased. Another convention to avoid conflicts with other packages is to define standard colors with a "the" prefix, as for example the color `blue` is defined as `the`⟨*blue*⟩. A similar convention is used for document elements. For example, the colors `thechaptercolor`, `thesectioncolor` and similar color definitions are used to color the title of a heading command.

E.6.1 Material Design

The naming conventions for these followaw the names provided by the authors of the scheme, colors are graded by numbers such as , , .

```
\def\palettename{palette}
```

Define the `glyphbox` color.

```
\definecolor{glyphbox}{rgb}{0.86,0.86,0.8}
%\definecolor{codebackground}{rgb}{0.8,0.8,1}
\definecolor{theblue} {rgb}{0.02,0.04,0.48}
\definecolor{thered}  {rgb}{0.65,0.04,0.07}
\definecolor{thegreen}{rgb}{0.06,0.44,0.08}
\definecolor{thelightgreen}{rgb}{0.06,0.44,0.06}
\definecolor{thegrey} {gray}{0.5}
\definecolor{thegray} {gray}{0.5}
\definecolor{thedarkgray} {gray}{0.95}
\definecolor{thelightgray}{gray}{0.6}


\colorlet{thefontname}{black}%font examples
```

Define the `thehighlight` color. This is used in conjunction with the `\hl`macro.

```
\colorlet{thehighlight}{yellow}%soul  highlight
```

Define the `thecancel` color.

```
\colorlet{thecancel}{thered}%for cancel commands
```

Define the `shadecolor` color.

```
\definecolor{shadedcolor}{gray}{0.6}
\definecolor{thelightgray}{gray}{0.6}
\definecolor{theshade}{gray}{0.94}
\definecolor{theframe}{gray}{0.75}
\definecolor{thecream}{rgb}{1,0.95,0.4}
\definecolor{spot}{rgb}{0,0.2,0.6}%some shades of blue
\definecolor{sweet}{rgb}{0,.68,.93}%shades of blue
%\colorlet{codebackground}{spot!5!white}
\definecolor{boxframe}{gray}{0.8}
\definecolor{boxfill}{rgb}{0.95,0.95,0.99}
\definecolor{theoption}{gray}{0.6}
\definecolor{ExampleFrame}{rgb}{0.628,0.705,0.942}
\definecolor{ExampleBack}{rgb}{0.963,0.971,0.994}
\colorlet{preciscolor}{sweet}
\definecolor{creamy}{HTML}{FDEBD7}
```

```
\definecolor{tofu}{HTML}{e7e3d8}
\colorlet{thetodo}{thegreen}
\newcommand{\TODO}{\textcolor{thetodo}{\bf TODO!}\xspace}
```

Once the colors are defined we are ready to create keys for the most common defined colors. **TODO!** (Fix me!)

```
\cxset{shadecolor/.code = \colorlet{shadecolor}{#1},
       thetodo/.code    = \colorlet{thetodo}{#1}}
\cxset{shadecolor = thecream,
       thetodo    = thetodo,}
```

Define the `shadecolor` color. Define the `thetodo` color.

E.7 Creating new palettes

`\createpalette`{⟨*name of color palette*⟩}{⟨*color space*⟩} {⟨*hex color code*⟩}{⟨*optional extra code*⟩}

Creates a palette, given a hex code and a name.

Following the design pattern in other sections we try to generalize the command. Given a color applicable to heading titles, I have set the color commands as best as I could. For some color combinations this is not satisfactory and modifications are necessary. This can be done by the `\addtocolorpalette` function.

```
\ExplSyntaxOn
\cs_gset:Npn \createpalette #1#2#3#4
  { \tl_gset:cn {auxpalette#1_tl} {}
    \addtotl {#1}{#2}{#3}{#4}
%
    \cxset{palette~#1/.code  = \cs:w auxpalette#1_tl \cs_end:
                               \edef\currentpalette{#1},

     }
    % the below are useless
    %\definecolor{palletteprimary}{#2}{#3}
    \definecolor{#1}{#2}{#3}
   % store the name of the palette
    \cs_gset:cpn {palette#1} {#1 }
 }
%
\cs_set:Npn \addtotl #1 #2 #3 #4
  {
    \tl_put_left:cn {auxpalette#1_tl}
      {
        \xcol_define_color:nnn {bgsexy}{#2}{#3}

        \xcol_define_color:nnn {thepalettecolor} {#2}{#3}
        \xcol_let:nn {thechaptercolor}{bgsexy}
        \xcol_let:nn{thechapternumbercolor}{bgsexy}
        \xcol_let:nn{thesectioncolor}{bgsexy}
        \xcol_let:nn{thesectiontitlecolor}{bgsexy}
        \xcol_let:nn{thesubsectioncolor}{bgsexy}
        \xcol_let:nn{thesubsectiontitlecolor}{bgsexy}
```

```
        \xcol_let:nn{thesubsubsectioncolor}{bgsexy}
        \xcol_let:nn{theparagraphcolor}{bgsexy}
        \xcol_let:nn{thesubparagraphcolor}{bgsexy}
        \xcol_let:nn{thesectionnumbercolor}{bgsexy}
        \xcol_let:nn{thesubsectionnumbercolor}{bgsexy}
        \xcol_let:nn{thesubsubsectionnumbercolor}{bgsexy}
        \xcol_let:nn{thelinkcolor}{bgsexy}
        \xcol_let:nn{thecommentstyle}{thegreen}
        \xcol_let:nn{thestringstyle}{thegreen}
        \xcol_let:nn{thekeywordstyle}{spot!90}
        \xcol_let:nn{theoperatorstyle}{thered}
        % code
        \xcol_let:nn{thecodebackground}{bgsexy!10}
        \xcol_let:nn{thecodeframe}{bgsexy!20}
        \xcol_let:nn{theoption}{black}
```

The following deal with color schemes for code highlighting in listings and documentation commands.

```
        \xcol_let:nn{theprimarycolor}{bgsexy}
        \xcol_let:nn{thecs}{bgsexy}
        \xcol_let:nn{theoption}{bgsexy}
        \xcol_let:nn{themodule}{bgsexy}
        %\xcol_let:nn{thecomment}{bgsexy}
        \xcol_let:nn{thebeginend}{bgsexy}
        \xcol_let:nn{theenv}{bgsexy}
        \xcol_let:nn{theenvironment}{bgsexy}
        \xcol_let:nn{thevalue}{bgsexy}
        \xcol_let:nn{thedoccommandcolor}{bgsexy}
        \xcol_let:nn{thekey}{bgsexy}
        \xcol_let:nn{theargument}{bgsexy}
        \xcol_let:nn{theoarg}{bgsexy}
        % maybe duplicate of the MANT consider removing.
        \xcol_let:nn{themeta}{thegreen}
        \xcol_let:nn{versionnote}{black!75}
        \xcol_let:nn{bibentry}{thegreen}
        \xcol_let:nn{bibentryfield}{black}
        \xcol_let:nn{theidx}{thegreen}
        % headers
        \xcol_let:nn{theplainoddheaderbgcolor}{white}
        \xcol_let:nn{theplainevenheaderbgcolor}{white}
        \xcol_let:nn{theplainoddfooterbgcolor}{white}
        \xcol_let:nn{theplainevenfooterbgcolor}{white}
        %
        \xcol_let:nn{theoddheaderbgcolor}{white}
        \xcol_let:nn{theevenheaderbgcolor}{white}
        \xcol_let:nn{theoddfooterbgcolor}{white}
        \xcol_let:nn{theevenfooterbgcolor}{white}
        %
        \xcol_let:nn{theheadingsoddheaderbgcolor}{white}
        \xcol_let:nn{theheadingsevenheaderbgcolor}{white}
        \xcol_let:nn{theheadingsoddfooterbgcolor}{white}
        \xcol_let:nn{theheadingsevenfooterbgcolor}{white}
        #4
        % rules
```

```
        \xcol_let:nn{theplainnoddheaderframerule}{bgsexy}
        \xcol_let:nn{theplainoddfooterframerule}{bgsexy}
        \xcol_let:nn{theplainevenheaderframerule}{bgsexy}
        \xcol_let:nn{theplainevenfooterframerule}{bgsexy}
        %
        \xcol_let:nn{theoddheaderframerule}{bgsexy}
        \xcol_let:nn{theoddfooterframerule}{bgsexy}
        \xcol_let:nn{theevenheaderframerule}{bgsexy}
        \xcol_let:nn{theevenfooterframerule}{bgsexy}
%
        \xcol_let:nn{theheadingsoddheaderframerule}{bgsexy}
        \xcol_let:nn{theheadingsoddfooterframerule}{bgsexy}
        \xcol_let:nn{theheadingsevenheaderframerule}{bgsexy}
        \xcol_let:nn{theheadingsevenfooterframerule}{bgsexy}
%     toc name color, i.e., Table of Contents
%     this is normally best to be black.
        \xcol_let:nn{thetocname}{bgsexy}
        \xcol_let:nn{thelofname}{bgsexy}
        \xcol_let:nn{thelotname}{bgsexy}
        \xcol_let:nn{thetocpart}{bgsexy}
        \xcol_let:nn{thetocchapter}{bgsexy}
        \xcol_let:nn{thetocsection}{black}
        \xcol_let:nn{thetocsubsection}{black}
%
        % list bullets
        \xcol_let:nn{theitemicolor}{bgsexy}
        \xcol_let:nn{theitemiicolor}{bgsexy}
        \xcol_let:nn{theitemiiicolor}{bgsexy}
        \xcol_let:nn{theitemivcolor}{bgsexy}
        \xcol_let:nn{theitemvcolor}{bgsexy}
        \xcol_let:nn{theitemvicolor}{bgsexy}
        % phd doc specific
        \xcol_let:nn{theunicodesymbolcolor}{bgsexy}
        \xcol_let:nn{thecmdcolor}{bgsexy}
        % epigraph rule
        \xcol_let:nn{theepigraphrulecolor}{bgsexy}
        %tabulars
        \xcol_let:nn{thetablevrulecolor}{bgsexy}
        \xcol_let:nn{thetableheadbgcolor}{bgsexy}

        %l3colors
        \xcol_let:nn { thebracescolor } { theblue }
        \xcol_let:nn { thesquarebracketscolor } { oprah }
        \xcol_let:nn { thecommentscolor } { green500 } %thepalettecolor does not look ve
        \xcol_let:nn { thecommentsstyle } { thepalettecolor }
        \xcol_let:nn { thecsmodulecolor } { unorange }
    }
}
```

We now create a number of color palettes for convenience. These are given various names, mostly from magazines or the web that have inspired the color schemes.

Define the `esquire` color.

```
\createpalette {esquire}     {HTML}  {D11C23} {}  % red shade
```

Define the fortune color.

```
\createpalette {fortune}      {HTML}  {EA8A4E} {}  % brick color
```

Define the oprah color.

```
\createpalette {oprah}        {HTML}  {F060A8} {}  % nice pinkish modern
```

Define the vogue color.

```
\createpalette {vogue}        {HTML}  {F21C93} {}  % nice pinkish modern
```

Define the bbc color.

```
\createpalette {bbc}           {HTML}  {991B1e} {}  % dark red
\tl_put_right:cn {auxpalettebbc_tl}
  {
    \colorlet{thecodebackground}{thelightgray!20}
  }
```

Define the architectural color.

```
\createpalette {architectural} {HTML} {0168FD} {}  % blue shade
```

Define the instyle

```
\createpalette {instyle}       {HTML} {227CE8} {}  % blue shade

```

Define the smithsonian

```
\createpalette {smithsonian}   {HTML} {60A8C0} {}  % milky blue
```

Define the blueprint color

```
\createpalette {blueprint}     {HTML} {486090} {}  % dark milky blue
```

Define the knoll color

E.7.1 Green Palettes

Define the oprahgreen color.

```
\createpalette {oprah~green}     {HTML} {6EC715} {}
\createpalette {oprahgreen}      {HTML} {6EC715} {}
\createpalette {knoll}         {HTML} {88A65E} {}  % nice sweet green
```

Define the living color

```
\createpalette {living}        {HTML} {678756} {}  % green shade
```

Define the spring onion

```
\createpalette {spring~onion}  {HTML} {90D228} {}  % bright green shade
\createpalette {springonion}  {HTML} {90D228} {}  % bright green shade
```

Define the olive color

```
\createpalette {olive}         {HTML} {EED38D} {}  % washed out not nice
```

Define the zealous color

```
\createpalette {zealous}       {HTML} {075D6B} {}  %
% orangey
\createpalette {orange~sakura} {HTML}{E6781E} {}  % nice modern serious book
\createpalette {orange}        {HTML}{FF6927} {}  % bright orange
\createpalette{unorange}       {HTML} {FE6B08} {}  %unbelievable orange
\createpalette{african~sun}    {HTML} {F76E2A} {}  %african sun http://www.colourlovers.
% brown
\createpalette {brown}         {HTML}{AF0C39} {}  % red brown
```

```
\createpalette {brown~red}     {HTML}{8D2420} {}  % brown red, serious luxury
% purples
\createpalette {black~tulip}   {HTML}{420943} {}  % darkish purple
\createpalette {helvetica}     {HTML}{404547} {}  % darkish purple
\createpalette {budapest}      {HTML}{BD7898} {}  % serious purple
%
\createpalette {cerulean}      {HTML}{9bb7d6} {}  % pantone light milky bluish
\createpalette {sealife }      {HTML}{7c7d89} {}  % sea life from https://www.benjaminmo
us/color-overview/color-collections/color-trends-2017
% reds
```

Define the `rouge` color.

```
\createpalette {rouge}         {HTML}{D2476F} {}  %rouge like in hot lips
\createpalette {sugar~hearts}  {HTML}  {FE4365} {}  % red rose http://www.colourlovers.c
\createpalette{Purple500}      {HTML} {7B1FA2} {}
%
%
\tl_put_right:cn {auxpalettecerulean_tl}
  {
    \colorlet{thecodebackground}{tofu}
  }
%
\tl_put_right:cn {auxpalettebluepprint_tl}
  {
    \colorlet{theunicodesymbolcolor}{bgsexy}
    \colorlet{thecodebackground}{thelightgray!20}
  }
\ExplSyntaxOff
\cxset{
/PLT
*PLT|DFLT
    % default palette
    palette orange sakura,
/PLT|DFLT
*PLT
}%
```

E.8 Hyperlinks

These colors are used for hyperlinking. We normally provide a uniform color for all hyperlinks

```
\colorlet{thelinkcolor}{bgsexy}   %linkcolor (also used in toc)
\colorlet{theanchorcolor}{bgsexy} %anchorcolor
\colorlet{thecitecolor}{bgsexy}   %citecolor
\colorlet{thefilecolor}{bgsexy}   %filecolor
\colorlet{themenucolor}{bgsexy}   %menucolor
\colorlet{theruncolor}{bgsexy}    %runcolor
\colorlet{theurlcolor}{bgsexy}    %urlcolor
\colorlet{Hyperlink}{bgsexy}      %{rgb}{0.281,0.275,0.485}
```

E.9 Code listings and documentation macros

For publications that require listings or self running examples we need to define a number of colors. We also need to cater for control sequences in the text and the like.

```
\definecolor{lstbgcolor}{rgb}{0.9,0.9,0.9}
\colorlet{examplefill}{yellow!80!black}
%
%\definecolor{thecodebackground}{HTML}{F2F2EA}
%\definecolor{thecodebackground}{HTML}{DED4B9}
%\definecolor{thecodebackground}{HTML}{F3EFE3} % light
\definecolor{thekeywordstyle}{HTML}{000000}
%\definecolor{thecommentstyle}{HTML}{DF8743}
\definecolor{thestringcolor}{HTML}{DF8743}
\definecolor{theemphasiscolor}{HTML}{DF8743}
\definecolor{theframerulecolor}{HTML}{FF6927}
\definecolor{theframefillcolor}{HTML}{000000}
\definecolor{theidentifiercolor}{HTML}{000000}
```

E.9.1 Specific code coloring for l3

The idea behind these colors is to provide a set of colors to colorize LaTeX 3 code, like in the implementation part of this document. LaTeX modules have distinct prefixes and sometimes suffixes, so it is unecessary to group them in long lists of keywords as listings does. My preference is to use softer colors and to colorize them in groups. Contrasting colors should be provided for braces, square brackets to ease reading.

Matching colors to the document palette main colors is not easy, especially if the code is typeset on a background.

```
% Color for braces, square brackets and comments.
% If you use the same color as headings, since the size and thickness
% are different your brain will interpet them as different colors!
```

```
\def\done{\cellcolor{teal}done}
\def\partialdone{\cellcolor{yellow}done}
\def\hcyan#1{{\color{teal} #1}}
```

E.10 User Commands

`\SaveCurrentPalette`

Saves the current palette into `\previouspalette`

```
\edef\SaveCurrentPalette{%
  \noexpand\edef\noexpand\previouspalette{\noexpand\currentpalette}}
```

The current default palette is the palette spring onion [2]

[2] 1.0,2013/01/26,Converted to DTX file.

Handlers Package Code Implementation Objectives and Strategy

The package provides macros for documenting and indexing primarily LaTeX or LaTeX 3 code. It combines the styles originitating with `doc` with those found in the `pgf` series of packages. It removes some of the difficulties of using hyperlinks in the index and elsewhere in the document. 21.00002

16

-2.5

-2.1

0.87865

> I was reflecting on the convoluted Java frameworks widely adopted at work. Those hefty frameworks brought coding structures and conventions to large engineering teams; meanwhile, they also sucked the fun of programming like a Pastafarian monster slurping all the tomato sauce on a plate of spaghetti.
>
> Zhen Wang

We start by outlining what we are trying to achieve with this package:

1. To provide a declarative interface to enable users to modify headings by setting keys, rather than writing macros.
2. The interface must be able to manupulate properties of headings down to the last detail.
3. To provide a compatibility mode, where documents wishing to test the package can have an easy switch to switch in and out. This is also important for the testing of the package.
4. To provide a number of templates that cover most of the typical use case.
5. To provide means for a plug-in architecture for extensions.

F.1 Terminology

document Any written item, as a book, article, or letter, especially of a factual or informative nature.

heading A division of a document or document series. For a normal book headings are chapters, sections etc. However we allow for specifying a more complex document divided into books, volumes parts etc. For example the Bible has Books, chapters and verses, where a legal document might require divisions such as clauses. In general these divisions are numbered. These document divisions are stored in the comma list **??** → P. ??.

head A typeset heading, such as chapter head, or section head. This can include a counter, label and title for example, *Chapter 1 Introduction*.

dom This is a programming interface that provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed. Although LaTeX does not offer a standard way to build such a tree (mainly because TeX does not require the marking of paragraphs, it is useful to think of the document as a tree structure. We also allow for a semi-automated way to build such a tree (with the exception that paragraphs are not included).

element A part of the document tree that can be styled on its own. For example the chapter label, or the section number.

F.2 Users

We classify users according to the LaTeX3 terminology as a) programmers b) template designers and c) authors.

F.2.1 Author

We assume that the author has an exising template which she is using but might want to do some minor modifications, for example use an italic shape for the font of the mark, but an upright font for the page numbers.

```
\cxset
 {
  chapter number color format          = apa,
  section title font-size font-size   = Large,
 }
```

We follow the idea of representing the basic elements of documents as elements, each one having a parent in order to specify the element we need to style as accurate as possible. One can think of this approach being congruent with objects in other languages. As a matter fact nothing stops us from defining a key value interface as shown below.

```
\cxset
 {
  header.even.mark.font.size = Large,
  header.even.mark.font.family = serif,
 }
```

This would pehaps make it easier for the template designer, but I have rejected the idea as my aim is to make it easy for the author, who can search the template and just enter a couple of new proerty values.

F.2.2 Template designer

The template designer in the example above would have selected the format style from a number of predefined formats (templates) or would have created a style called *apa* from an existing template and modified it using declarative key style.

F.2.3 The programmer

The programmer in the example above could have created the basic format *apa* by using both declarative as well as defining or using existing macros. To the programmer we offer an extension mechanism, where the contents of a `ps@` command are defined. For example the programmer can define a new style using Ti*k*Z, but without having to worry about defining full `ps@` and their interface.

F.3 Preliminaries

Standard file identification. We first announce the package and require that it be used with LaTeX2e.

```
\NeedsTeXFormat{LaTeX2e}[2018/04/15]%
\ProvidesPackage{phd-handlers}[2015/1/13 v1.0 less preamble (YL)]%
\g@addto@macro \theindex { \MakePrivateLetters }
```

F.4 Source2e Interface

I am not very fond of mixing expl3 control sequences with source2e commands. Here we provide an interface for all these commands we might use. This section can be revisited once expl3 code becomes available.

```
\ExplSyntaxOn
\let\ltxtoday\today
\let\phd_hang_from:nn \@hangfrom
\newif\if@ltxcompat \@ltxcompatfalse
\ExplSyntaxOff
```

F.5 PGF Key Handlers

One of the greateset ideas in key management was pgf's handlers. New handlers can be added in quite easily and they are very versatile in this respect.

F.5.1 Font family handler

This handler first checks if the font-family has been supplied as a list. In this case it will call check font and pick the first available font.

```
\ExplSyntaxOn
\pgfkeys{%
   /handlers/.alias/.code=
      \pgfkeysedef\pgfkeyscurrentpath{%
                  \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}},%
     % /handlers/.alias/.value required,%
     % /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
     % /handlers/.blank/.default=\pgfkeysnovalue,%
}

\clist_new:N \allowed_font_family_options
\clist_gset:Nn \allowed_font_family_options
  {
    sffamily,rmfamily,ttfamily, normal,italic,oblique,
    itshape,slshape,upshape,sans-serif,serif,monospace,
    inherit,auto,normalfont
  }
% create aliases for some of the names to match those commonly
% used in css  NEEDS TESTING
\cs_gset_eq:cc {bold} {bfseries}

```

```
  \pgfkeys{/handlers/.font-family~in/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \clist_if_in:NnTF \allowed_font_family_options {##1 }
          {
           \gdef #1 {##1}
          }
          {
            \gdef #1 {\sffamily}
          }
      }
  }
   \pgfkeys{/handlers/.fontfamily/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { none       } { \cs_gset:cpn {#1} { \normalfont } }
               { rmfamily   } { \cs_gset:cpn {#1} { \rmfamily   } }
               { serif      } { \cs_gset:cpn {#1} { \rmfamily   } }
               { sans-serif } { \cs_gset:cpn {#1} { \sffamily   } }
               { sffamily   } { \cs_gset:cpn {#1} { \sffamily   } }
               { ttfamily   } { \cs_gset:cpn {#1} { \ttfamily   } }
              }
             {                                    }
             { \cs_gset:cpn {#1} {\cs:w ##1 \cs_end:} }
     }
   }

\ExplSyntaxOff
```

F.5.2 Font face handler

The handler `.font-face in` creates a command for activating a font face and also activates if it was not ?

`\title_font_face` ⟨*font name*⟩

This peculiar handler is a store in that can be used for all elements. We defined them as ⟨*element*⟩`_font_face`. This took a bit long to figure out. The `allowed_font_face_options` will eventually be linked with the font loading mechanism.

```
\ExplSyntaxOn
\clist_new:N \allowed_font_face_options
\clist_gset:Nn \allowed_font_face_options
  {
    sffamily,rmfamily,ttfamily, normal,italic,oblique,
    itshape,slshape,upshape,sans-serif,serif,monospace,
    inherit,auto,tiresias,arial,pan
  }
% creat aliases for some of the names to match those commonly
% used in css.
```

```

 \pgfkeys{/handlers/.font-face~in/.code =
   \pgfkeysalso
     {\pgfkeyscurrentpath/.code=
        \IfNoValueTF {##1} {\@empty}
          {
           \cs_if_exist:NTF { #1 }
             { \gdef#1 {##1} }
             { \gdef#1 {\rmfamily} } %add error trapping as well
          }
         }

 }

\ExplSyntaxOff
```

F.5.3 Font face handler

```
\ExplSyntaxOn
 \pgfkeys{/handlers/.set~font-face/.code=\pgfkeysalso {
     \pgfkeyscurrentpath/.code=
           \def\tempa{##1}
           \def\tempb{normal}
           \def#1{##1}
           \ifx\tempa\tempb
              \def#1{\normalfont}
           \fi
   }
 }
\ExplSyntaxOff
```

F.5.4 Fonts style handler

This controls the input for font-shape or font-style, which they are equivalent.

```
\ExplSyntaxOn
\clist_new:N \allowed_font_style_options
\clist_gset:Nn \allowed_font_style_options {normal,upshape,itshape,oblique,slshape,ital
% create some aliases also for these options.
\cs_gset_eq:cc {normal}{upshape}
\cs_gset_eq:cc {italic}{itshape}
\cs_gset_eq:cc {oblique}{slshape}

\pgfkeys{/handlers/.font-style~in/.code =

    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \clist_if_in:NnTF \allowed_font_style_options {##1 }
          {
           \gdef #1 {##1}
          }
          {
            \gdef#1 {upshape} %Plus error message
          }
```

```
      }
  }
  \pgfkeys{/handlers/.fontstyle/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { normal       } { \cs_gset:cpn {#1} { \upshape   } }
               { upshape      } { \cs_gset:cpn {#1} { \upshape   } }
               { itshape      } { \cs_gset:cpn {#1} { \itshape   } }
               { italic       } { \cs_gset:cpn {#1} { \itshape   } }
               { oblique      } { \cs_gset:cpn {#1} { \slshape   } }
               { slshape      } { \cs_gset:cpn {#1} { \slshape   } }
               { scshape      } { \cs_gset:cpn (#1) { \scshape   } }
             }
             {                                                      }
             { \cs_gset:cpn {#1} {##1}                          }
      }
   }

\pgfkeys{/handlers/.listformat/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { normal       } { \cs_gset:cpn {#1} { \upshape   } }
               { upshape      } { \cs_gset:cpn {#1} { \upshape   } }
               { itshape      } { \cs_gset:cpn {#1} { \itshape   } }
               { italic       } { \cs_gset:cpn {#1} { \itshape   } }
               { oblique      } { \cs_gset:cpn {#1} { \slshape   } }
               { slshape      } { \cs_gset:cpn {#1} { \slshape   } }
               { scshape      } { \cs_gset:cpn (#1) { \scshape   } }
             }
             {                                                      }
             { \cs_gset:cpn {#1} {##1}                          }
      }
   }
\ExplSyntaxOff
```

`\title_font_weight` ⟨*font weight name*⟩

Handler for defining the font-weight of an element.

These are all very repetitive and could have been replaced by one handler at the cost of obfuscating the code.

```
\ExplSyntaxOn
\clist_new:N \allowed_font_weight_options
\clist_gset:Nn \allowed_font_weight_options
  {
    normal,bold,bfseries,mdseries,none
  }
 \pgfkeys{/handlers/.font-weight~in/.code =
    \pgfkeysalso
```

```
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { none        } { \cs_gset:Npn #1 { empty    } }
               { bold        } { \cs_gset:Npn #1 { bfseries } }
               { normal      } { \cs_gset:Npn #1 { normalfont } }
               { bfseries    } { \cs_gset:Npn #1 { bfseries } }
               { mdseries    } { \cs_gset:Npn #1 { mdseries } }
              }
             {                          }
             { \cs_gset:Npn #1 {empty} }
      }
}
```

The `.fontweight` handler is used to define keys for `font-weight`.

```
   \pgfkeys{/handlers/.fontweight/.code =
   \pgfkeysalso
     {\pgfkeyscurrentpath/.code=
       \tl_set:Nn\l_tmpa_str:N {##1}
          \str_case_x:nnTF {##1}
            {
              { none       } { \cs_gset:cpn {#1} { } }
              { bold       } { \cs_gset:cpn {#1} { \bfseries } }
              { normal     } { \cs_gset:cpn {#1} { } }
              { bfseries   } { \cs_gset:cpn {#1} { \bfseries } }
              { mdseries   } { \cs_gset:cpn {#1} { \mdseries } }
             }
            {                                }
            {  \cs_gset:cpn {#1} { }         }
     }
   }
\ExplSyntaxOff
```

The `font-size in` handler is defined next. This can be set both as a command or a name.

`\allowed_keys_font_size`⟨*list of allowed keys*⟩

list with all allowed keys in the **`phd system`**

Coding this through **`expl3`** is such a pleasure and cut the previous code in half! The mavericks of the LaTeX3 Team pulled it off! It also enables better error trapping and the allowed words can be configurable, one day! For example if 38/52 is typed one can automatically set it to a specific size? Could also allow say 10pt check in list if we have them and set it. The keys auto, inherit, autoscale need development.

`auto` should scale one step down from parent element. **`inherit`** takes the same value as the parent element **`autoscale`** will fit the contents in the width of its container box. (All three for next version)

```
\ExplSyntaxOn

  \clist_new:N   \allowed_keys_font_size
  \clist_gset:Nn \allowed_keys_font_size
    {
```

```
      normal,tiny,footnotesize,small,
      scriptsize,
      large,Large,LARGE,
      huge,Huge,HUGE,HHUGE,
      HHHUGE,auto,inherit,
      autoscale,
    }

  \pgfkeys{/handlers/.font-size~in/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \clist_if_in:NnTF \allowed_keys_font_size {##1 }
          {
           \cs_gset:Npn #1  {##1}
          } {normalsize}
      }
   }
```

When the storing key is provided as a name we use the `.fontsize` handler.

```
    \pgfkeys{/handlers/.fontsize/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { normal     } { \cs_gset:cpn {#1} { \normalsize } }
               { tiny      } { \cs_gset:cpn {#1} { \tiny      } }
               { small      } { \cs_gset:cpn {#1} { \small      } }
               { scriptsize } { \cs_gset:cpn {#1} { \scriptsize } }
               { large      } { \cs_gset:cpn {#1} { \large      } }
               { Large      } { \cs_gset:cpn {#1} { \Large      } }
               { LARGE      } { \cs_gset:cpn {#1} { \LARGE      } }
               { huge       } { \cs_gset:cpn {#1} { \huge       } }
               { Huge       } { \cs_gset:cpn {#1} { \huge       } }
               { HUGE       } { \cs_gset:cpn {#1} { \HUGE       } }
               { HHUGE      } { \cs_gset:cpn {#1} { \HHUGE      } }
               { HHHUGE     } { \cs_gset:cpn {#1} { \HHHUGE     } }
              }
             {                          }
             { \cs_gset:cpn {#1} {##1} }
      }
   }
\ExplSyntaxOff
```

F.5.5 Capitalization handler

This handler is for making text upper or lower case. Took a few tests to get it right! Will have to go to expl3 when these work ok. Fire

```
\ExplSyntaxOn
  \clist_new:N  \allowed_options_case
  \clist_gset:Nn \allowed_options_case
    {
      lower,upper,none
```

```
    }

  \pgfkeys{/handlers/.case~in/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { none       } { \cs_gset:Npn #1 { empty             } }
               { lower      } { \cs_gset:Npn #1 { MakeTextLowercase } }
               { lowercase  } { \cs_gset:Npn #1 { MakeTextLowercase } }
               { upper      } { \cs_gset:Npn #1 { MakeTextUppercase } }
               { uppercase  } { \cs_gset:Npn #1 { MakeTextUppercase } }
               { upper case } { \cs_gset:Npn #1 { MakeTextUppercase } }
             }
             {                          }
             { \cs_gset:Npn #1 {empty} }
      }
  }

\pgfkeys
 {/handlers/.case/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
         \str_case_x:nnTF {##1}
             {
               { none       } { \cs_gset:cpn {#1} { \empty             } }
               { lower      } { \cs_gset:cpn {#1} { \MakeTextLowercase } }
               { lowercase  } { \cs_gset:cpn {#1} { \MakeTextLowercase } }
               { upper      } { \cs_gset:cpn {#1} { \MakeTextUppercase } }
               { uppercase  } { \cs_gset:cpn {#1} { \MakeTextUppercase } }
               { upper~case } { \cs_gset:cpn {#1} { \MakeTextUppercase } }
             }
             {                          }
             { \cs_gset:cpn #1 {\MakeTextLowercase} }
      }
  }

 \pgfkeys
 {/handlers/.store/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
         \cs_gset:cpn {#1} {##1}
      }
  }
\ExplSyntaxOff
```

F.5.6 Alignment handler

This handler is for aligning text. We cater for all variations.

```
\ExplSyntaxOn
  \pgfkeys{/handlers/.textalign/.code =
    \pgfkeysalso
```

```
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { none        } {  \cs_gset:cpn {#1} {}                }
               { left        } {  \cs_gset:cpn {#1} { \RaggedRight } }
               { RaggedLeft  } {  \cs_gset:cpn {#1} { \RaggedLeft  } }
               { raggedleft  } {  \cs_gset:cpn {#1}  { \raggedleft } }
               { right       } {  \cs_gset:cpn {#1} { \RaggedLeft  } }
               { RaggedRight } {  \cs_gset:cpn {#1} { \RaggedRight } }
               { raggedright } {  \cs_gset:cpn {#1} { \raggedright } }
               { center      } {  \cs_gset:cpn {#1} { \Centering   } }
               { centre      } {  \cs_gset:cpn {#1} { \Centering   } }
               { centering   } {  \cs_gset:cpn {#1} { \centering   } }
               { Centering   } {  \cs_gset:cpn {#1} { \Centering   } }
               { flushleft   } {  \cs_gset:cpn {#1} { \flushleft   } }
               { flushright  } {  \cs_gset:cpn {#1} { \flushright  } }
             }
             {                           }
             { \cs_gset:cpn #1 {\empty} }
      }
  }
\ExplSyntaxOff
```

F.5.7 On off general handlers

This handler is for programmatically setting on off variables.

```
\ExplSyntaxOn
  \pgfkeys{/handlers/.onoff/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { on     }  { %\cs_gset:cpn {#1} {##1}
                               \bool_if_exist:cTF {#1} { \bool_gset_true:c {#1} }
                                                    { \bool_new:c {#1}
                                                      \bool_gset_true:c {#1}
                                                    }
                            }

               { off    }  {
                               \bool_if_exist:cTF {#1} { \bool_gset_false:c {#1} }
                                                    { \bool_new:c {#1}
                                                      \bool_gset_false:c {#1}
                                                    }
                            }

              { true    }  { %\cs_gset:cpn {#1} {##1}
                               \bool_if_exist:cTF {#1} { \bool_gset_true:c {#1} }
                                                    { \bool_new:c {#1}
                                                      \bool_gset_true:c {#1}
                                                    }
```

```
                            }

                { false    }  {
                                \bool_if_exist:cTF {#1} { \bool_gset_false:c {#1} }
                                                    { \bool_new:c {#1}
                                                      \bool_gset_false:c {#1}
                                                    }
                            }


               }
               {                            }
               { }%\cs_gset:cpn {#1} {empty} }
        }
   }
\ExplSyntaxOff
```

F.5.8 Shadow handlers

Many components can be rendered with shadows. This can be done through the tcolorbox shadowing commands or directly through Ti*k*Z. The way parameters are specified in both cases results in multi-argument keys, which is generally against the philosophy of the mark-up semantics of the **phd** package. Most of these keys required color specification and or size specification.

Since colors are linked to palettes, I decided that the color part would belong to the palette settings rather than the shadow keys. Also any sizing of shadows has been delegated to default macros. This simplifies the user interface tremendously. Should a template designer wish to provide a more complicated shadow, this can be achieved through the style property of the element.

Shadows are only available when the box rendering engine depend on tcolorbox.

```

\ExplSyntaxOn
 \pgfkeys
 {/handlers/.shadow/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
         \str_case_x:nnTF {##1}
             {
               { none                } { \cs_gset:cpn {#1} { {no~shadow} } }
               { off                } { \cs_gset:cpn {#1} { {no~shadow} } }
               { on                } { \cs_gset:cpn {#1} { {drop~shadow} } }
               { drop~shadow        } { \cs_gset:cpn {#1} { drop~shadow } }
               { drop~lifted~shadow } { \cs_gset:cpn {#1} { {##1}       } }
               { fuzzy~halo         } { \cs_gset:cpn {#1} { halostyle   } }
             }
             {                       }
             { \cs_gset:cpn {#1} {##1} }
        }
   }
\ExplSyntaxOff
```

31/07/2017 Nasty bug fix, did not brace one argument of cpn, resulted redefining the l

F.5.9 Sectioning format handler

This handler manages sectioning type formats. Not all of them are currently defined.

```
\ExplSyntaxOn
  \msg_new:nnnn {phd-handlers} {undefined-format}
    {`#1'~is~not~a~defined~format}
    {You~have~to~define~and~register~the~format~`#1'~before~you~can~use~it.}

  \clist_new:N   \allowed_options_format
  \clist_gset:Nn \allowed_options_format
    {
      plain, block, display, runin, leftmargin,
      rightmargin, drop, wrap, frame, custom, inline, inmargin,
      hanginmargin, paperback,fashion,traditional,hang,stewart,
    }
  \pgfkeys{/handlers/.format~in/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \clist_if_in:NnTF \allowed_options_format {##1 }
          {
           \cs_gset:cpn {#1} {##1} %true code
          }
          {
             \cs_if_exist:cTF{##1}{ \cs_gset:cpn {#1}  {##1}}
             {
               \msg_error:nnx { phd-handlers } { undefined-format } {##1}
               % define it in any case until we have a stable API
               \cs_gset:cpn {#1}  {##1}
             } %false code
          } % to add that this is custom message and check if defined
      }
  }
\ExplSyntaxOff

\ExplSyntaxOn
   \msg_new:nnnn {phd-handlers} {undefined-color}
    {`#1'~is~not~a~defined~color}
    {You~have~to~define~the~color~`#1'~before~you~can~use~it.~I~am~using~the~`current~de
  \clist_new:N   \allowed_color_options
  \clist_gset:Nn \allowed_color_options
    {
      auto, palette,
    }
  \pgfkeys{/handlers/.colorin/.code =
    \pgfkeysalso
      {\pgfkeyscurrentpath/.code=
        \clist_if_in:NnTF \allowed_color_options  {##1 }
          {
```

```
            %check to see if palette is defined
            \cs_if_exist:cTF{palettename}{\cs_gset:cpn{#1} {\color{the#1color}} }{
           % \cs_gset:cpn {#1} {\color{##1}}
            }

           }
           {
            {
            % We check that the color is defined
               \@ifundefined{\string\color@##1}{
                  \msg_error:nnx { phd-handlers } { undefined-
color } {##1}
                  \colorlet{##1}{black}
                }
                {
                  \cs_gset:cpn {#1} {\color{##1}}
                }
            } %FIRE MUST CATER FOR SPECIALS
           } % to add that this is custom message and check if defined
       }
  }
\ExplSyntaxOff
```

F.5.10 Number format handler

This handler handles the definition of **\the**⟨*section*⟩ type commands. As the first argument of the key handler is a command we need to remove the **\the** part in order to define the counter correctly. One day we will remove all LaTeX's counters and use our own, then thsi would be unecessary. as we need to gobble four tokens we need to define a **\use_v:nnnnn**, as **expl3** does not have one.

\use_v:nnnnn {⟨*arg1*⟩} {⟨*arg2*⟩} {⟨*arg3*⟩} {⟨*arg4*⟩} {⟨*arg5*⟩}

Gobbles five arguments.

```
\ExplSyntaxOn
  \cs_set:Npn \use_v:nnnnn #1#2#3#4#5 {#5}
   \cs_new:Npn \remove_the:#1
     {
       \exp_after:wN \use_v:nnnnn \token_to_str:N #1
     }
```

next we provide a function to remake the **the<name>** based on the key handler input.

```
   \cs_gset:Npn \make_the_cmd: #1#2
     {
       \cs_gset:Npn \countername { \remove_the:{#1} }
       \cs_gset:Npn #1
         {
           \exp_after:wN \exp_after:wN \exp_after:wN #2
           \exp_after:wN \cs:w c@\countername\cs_end:
         }
    }
```

In comparison to the mind twister above the key-handler is easily defined next and cuts quite a bit of code later.

```
  \pgfkeys{/handlers/.numbering~in/.code =

    \pgfkeysalso
      {
        \pgfkeyscurrentpath/.code=
        \tl_set:Nn\l_tmpa_str:N {##1}
           \str_case_x:nnTF {##1}
             {
               { none       } { \cs_gset:Npn #1     {}                }
               { arabic     } { \make_the_cmd: {#1} { \@arabic     } }
               { Alpha      } { \make_the_cmd: {#1} { \@Alph       } }
               { alph       } { \make_the_cmd: {#1} { \@alph       } }
               { Roman      } { \make_the_cmd: {#1} { \@Roman      } }
               { roman      } { \make_the_cmd: {#1} { \@roman      } }
               { WORDS      } { \make_the_cmd: {#1} { \WORDS@cx    } }
               { Words      } { \make_the_cmd: {#1} { \Words@cx    } }
               { words      } { \make_the_cmd: {#1} { \words@cx    } }
               { ordinals   } { \make_the_cmd: {#1} { \ordinals@cx } }
               { Ordinals   } { \make_the_cmd: {#1} { \ordinals@cx } }
               { ORDINALS   } { \make_the_cmd: {#1} { \ORDINALS@cx } }
               % add padzeroes later
             }
             {                          }
             { \cs_gset:Npn #1 {##1}    }
      }
  }
\ExplSyntaxOff
```

</HANDLER>[1]

[1] 1.0,2013/01/26,Converted to DTX file.

IMPLEMENTATION AND FRIENDS

phd package and give a background as to how fonts are used in LaTeX.

phd-sect Code Implementation Objectives and Strategy

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

> I was reflecting on the convoluted Java frameworks widely adopted at work. Those hefty frameworks brought coding structures and conventions to large engineering teams; meanwhile, they also sucked the fun of programming like a Pastafarian monster slurping all the tomato sauce on a plate of spaghetti.
>
> Zhen Wang

We start by outlining what we are trying to achieve with this package:

1. To provide a declarative interface to enable users to modify headings by setting keys, rather than writing macros.
2. The interface must be able to manupulate properties of headings down to the last detail.
3. To provide a compatibility mode, where documents wishing to test the package can have an easy switch to switch in and out. This is also important for the testing of the package.
4. To provide a number of templates that cover most of the typical use case.
5. To provide means for a plug-in architecture for extensions.

G.1 Terminology

document Any written item, as a book, article, or letter, especially of a factual or informative nature.

heading A division of a document or document series. For a normal book headings are chapters, sections etc. However we allow for specifying a more complex document divided into books, volumes parts etc. For example the Bible has Books, chapters and verses, where a legal document might require divisions such as clauses. In general these divisions are numbered. These document divisions are stored in the comma list `phd_book_divisions_clist`.

head A typeset heading, such as chapter head, or section head. This can include a counter, label and title for example, *Chapter 1 Introduction*.

dom This is a programming interface that provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed. Although LaTeX does not offer a standard way to build such a tree (mainly because TeX does not require the marking of paragraphs, it is useful to think of the document as a tree structure. We also allow for a semi-automated way to build such a tree (with the exception that paragraphs are not included).

element A part of the document tree that can be styled on its own. For example the chapter label, or the section number.

G.2 Users

We classify users according to the LaTeX3 terminology as a) programmers b) template designers and c) authors.

G.2.1 Author

We assume that the author has an exising template which she is using but might want to do some minor modifications, for example use an italic shape for the font of the mark, but an upright font for the page numbers.

```
\cxset
 {
  chapter number color format          = apa,
  section title font-size font-size   = Large,
 }
```

We follow the idea of representing the basic elements of documents as elements, each one having a parent in order to specify the element we need to style as accurate as possible. One can think of this approach being congruent with objects in other languages. As a matter fact nothing stops us from defining a key value interface as shown below.

```
\cxset
 {
  header.even.mark.font.size = Large,
  header.even.mark.font.family = serif,
 }
```

This would pehaps make it easier for the template designer, but I have rejected the idea as my aim is to make it easy for the author, who can search the template and just enter a couple of new proerty values.

G.2.2 Template designer

The template designer in the example above would have selected the format style from a number of predefined formats (templates) or would have created a style called *apa* from an existing template and modified it using declarative key style.

G.2.3 The programmer

The programmer in the example above could have created the basic format *apa* by using both declarative as well as defining or using existing macros. To the programmer we offer an extension mechanism, where the contents of a `ps@` command are defined. For example the programmer can define a new style using Ti*k*Z, but without having to worry about defining full `ps@` and their interface.

G.2.4 The publisher

Occassionally the publisher might need further macros and keys to assist with revisions, internal tracking and collation of manuscripts from different authors.

G.3 Preliminaries

Standard file identification. We first announce the package and require that it be used with LaTeX2e.

```
@@=phd
```

```
\NeedsTeXFormat{LaTeX2e}[2017/04/15]%
\ProvidesFile{phd-lowersections}[2017/04/15 v1.0 Better headings (YL)]%
```

G.4 Source2e Interface

I am not very fond of mixing expl3 control sequences with source2e commands. Here we provide an interface for all these commands we might use. This section can be revisited once expl3 code becomes available.

```
\ExplSyntaxOn
\let\ltxtoday\today
\let\phd_hang_from:nn \@hangfrom
\newif\if@ltxcompat \@ltxcompatfalse
\ExplSyntaxOff

%\newcommand\tikzi[1][]{}

\newif\if@DEBUG
\@DEBUGfalse



\def\DEBUGON{
\@DEBUGtrue
\if@DEBUG
\def\tikzi[##1]{%
  \begin{tikzpicture}[remember picture,overlay]
    \draw[<->] (0,0)--(0,.2)--++(-.5,0) node[left,fill=blue!15,text=black]%
      {
        \parbox{2cm}
        {
          {\sffamily
           \footnotesize ##1 }
         }
        };
  \end{tikzpicture}%
}
\else
\def\tikzi[##1]{}
\fi}
```

```


\def\DEBUGOFF{
\@DEBUGfalse
\if@DEBUG
\def\tikzi[##1]{%
    \begin{tikzpicture}[remember picture,overlay]
                \draw[<->] (0,0)--(0,.2)--++(-.5,0) node[left,fill=blue!5,text=black]%
                {
                   \parbox{2cm}
                    {
                       {\sffamily
                        \footnotesize ##1 }
                    }
                };
   \end{tikzpicture}%
}
\else
\def\tikzi[##1]{}
\fi
}

\DEBUGON

```

G.5 Key Management

This part of the code is a bit verbose. We care to provide keys for all parameters in order to allow flexibility and easy extensions.

```
\ExplSyntaxOn
\def\lphdsubsectionnumberprefixtl {}
\def\lphdsubsectionnumbersuffixtl {}
\def\lphdsubsubsectionnumberprefixtl{}
\def\lphdsubsubsectionnumbersuffixtl{}
\def\lphdparagraphnumberprefixtl{}
\def\lphdparagraphnumbersuffixtl{}
\def\lphdsubparagraphnumberprefixtl{}
\def\lphdsubparagraphnumbersuffixtl{}
```

\phd_document_type_tl{⟨*type of document*⟩}

A macro holding the literal type name of the document, i.e, article, proceedings, book, letter etc. This will be used to enable user commands to be created for maximum flexibility.

\@@_document_types_clist{⟨*list of types provided*⟩}

An associated clist that holds all the document types available by the class.

```
\clist_new:N \__phd_document_types_clist {}
```

The build-in default document types are those provided by the standard classes, book, article etc.

```
\tl_new:N \__phd_document_type_tl {}
\cs_new_nopar:Npn \__phd_set_document_type:n #1
  {
    \tl_gset:Nn \__phd_document_type_tl {#1}
    %create secondary list as well here
    \clist_gput_right:Nn \__phd_document_types_clist {#1}
    \clist_new:c {__phd_#1_divisions_clist}
  }
```

We set the default document type as *book*.

```
\__phd_set_document_type:n {book}
```

\phd_book_divisions_clist {⟨*clist*⟩}

This clist contains all the book divisions, we later also provide an interface for the user to add new content to it. For example this document is of type book.

```
\cs_new:Npn \__phd_set_document_divisions #1 #2 {
  \clist_gset:cn {__phd_#1_divisions_clist} {#2}
}

\__phd_set_document_divisions {book}
  {
    part,chapter,section,subsection,subsubsection,
    paragraph,subparagraph
  }

\__phd_set_document_divisions {article}
  {
    section,subsection,subsubsection,
    paragraph,subparagraph
  }

\__phd_set_document_divisions {proc}
  {
    section,subsection,subsubsection,
    paragraph,subparagraph
  }

\__phd_set_document_divisions {journal}
  {
    volume,article,section,subsection,subsubsection,
    paragraph,subparagraph
  }
%
```

Somehow controversial, we divide a bible-like book into testament, book, chapter and verse. As we can create new ones to one's liking, we leave other religious main body subdivisions for another time.

```
\__phd_set_document_divisions {bible}
  {
    testament,book,chapter,verse
  }
```

The structural elements of the main body of a publication, have been dealt with the code described so far. For dealing with the visual parts for such elements we have to define, subdivisions to cater for each one of them.

\phd_secondary_clist{⟨*clist*⟩}

Holds a list of all secondary elements for a book. Each main element has an associated number of secondary elements, which are used in typesetting the element. Most sections will have a title, others will have an associated counter with them.

```
\cs_new:Npn \element_delimiter{.}
\clist_new:N \thirdlist
\cs_new:Npn \z_make_names #1 {
   % append to third list
   \clist_map_inline:Nn \__phd_book_divisions_clist {
     \clist_put_right:Nn \thirdlist{##1\element_delimiter#1}
   }
 }

\clist_new:N   \__phd_secondary_clist_names_clist
\clist_gset:Nn \__phd_secondary_clist_names_clist
 {
   label,label_before,number,title, title_before,number_before
 }
% map the two lists and create a third
\clist_map_inline:Nn \__phd_secondary_clist_names_clist
  {
     \z_make_names {#1}
  }
```

Testing everything so far, we get a mind-boggling list of sub-elements that we can provide keys and settings for them. Each one of them will have numerous properties associated with them.

part.label,
chapter.label,
section.label,
subsection.label,
subsubsection.label,
paragraph.label,
subparagraph.label,
part.label_before,
chapter.label_before,
section.label_before,
subsection.label_before,
subsubsection.label_before,
paragraph.label_before,
subparagraph.label_before,
part.number,
chapter.number,
section.number,
subsection.number,
subsubsection.number,
paragraph.number,
subparagraph.number,
part.title,
chapter.title,
section.title,
subsection.title,
subsubsection.title,
paragraph.title,
subparagraph.title,
part.title_before,
chapter.title_before,
section.title_before,
subsection.title_before,
subsubsection.title_before,
paragraph.title_before,
subparagraph.title_before,
part.number_before,

chapter.number_before,
section.number_before,
subsection.number_before,
subsubsection.number_before,
paragraph.number_before,
subparagraph.number_before

```
\clist_new:N \__phd_secondary_clist
\clist_gset:Nn \__phd_secondary_clist
{
  part_label,
  part_label_before,
  part_number,
  part_title,
  part_title_before
  part_number_before,

  chapter_title,
  section_title,
  subsection_title,
  subsubsection_title,
  paragraph_title,
  subparagraph_title,
  chapter_title_before,
%
  chapter_label,
  section_label,
  subsection_label,
  subsubsection_label,
  paragraph_label,
  subparagraph_label,
  chapter_label_before,
%
  chapter_number,section_number,
  subsection_number,subsubsection_number,
  paragraph_number,subparagraph_number,

  chapter_number_before,

}
```

\phd_create_new_element:nn {⟨*label*⟩}

Creates new dims for primary elements. The label is one of chapter, section etc.

```
\cs_new:Npn \__phd_create_new_element:nn #1
  {
    \dim_gzero_new:c {l__phd_#1_margin_top_width_dim} %margin
    \dim_gzero_new:c {l__phd_#1_margin_right_width_dim}
    \dim_gzero_new:c {l__phd_#1_margin_bottom_width_dim}
    \dim_gzero_new:c {l__phd_#1_margin_left_width_dim}

    \dim_gzero_new:c {l__phd_#1_padding_top_width_dim} %margin
    \dim_gzero_new:c {l__phd_#1_padding_right_width_dim}
    \dim_gzero_new:c {l__phd_#1_padding_bottom_width_dim}
    \dim_gzero_new:c {l__phd_#1_padding_left_width_dim}

```

```
    \dim_gzero_new:c {l__phd_#1_border_top_width_dim} %margin
    \dim_gzero_new:c {l__phd_#1_border_right_width_dim}
    \dim_gzero_new:c {l__phd_#1_border_bottom_width_dim}
    \dim_gzero_new:c {l__phd_#1_border_left_width_dim}
 }
```

Next we create the secondary elements.

```
\clist_map_inline:Nn \__phd_book_divisions_clist
  {
    \__phd_create_new_element:nn {#1}
  }



\clist_map_inline:Nn \__phd_secondary_clist
  {
    \__phd_create_new_element:nn {#1}
  }


\cs_new:Npn \printproperties
  {
    \input{debug-dims}
  }
\ExplSyntaxOff
```

`\l_@@_make_new_heading_keys:n` ⟨*element name*⟩

Creates a new set of keys for an element for exmaple, chapter, section or chapter title. Not all keys are applicable to all elements (for example numbering), hence the code is split into primary and secondary elements. A primary element is an element whose parent is **`document`**.

```
\ExplSyntaxOn
\cs_new:Npn \l__phd_make_new_heading_keys:n #1 #2
 {
  \cxset
    {
```

The first key we define is to set the element name. This is compatible with all classes, e.g. `\chaptername` () for the name of the chapter, `\sectionname` (Section) etc. The handler **`.store`** is similar to PGF's **`.store in`**, except it accepts a **`csname`**.

```
      #1~name/.store                = #1name,
```

At this point we should think as to how we can internationalize our code. We create a path **`/phd/locale/en/chaptername`** to store the various names. This will be extended later to cater for other aspects of internationalization.

```
      locale/en/#1~name/.store      = #1name,
```

We avoid having to load packages to set the indentation of the first paragraph after a heading on or off. The handler **`.onoff`** is defined in the phd-handlers package and can take a value of on, off, true, false. As we use expl3 we get a value of **`\char"0`** for false or off.

```
    #1~afterindent/.onoff          = #1_afterindent,
% Main text alignment
    #1~align/.textalign            = l__phd_#2_align_tl,
% spacing
    #1~beforeskip/.store           = l__phd_#2_beforeskip_tl,
    #1~afterskip/.store            = l__phd_#2_afterskip_tl,
    #1~indent/.store               = l__phd_#2_indent_tl,
```

The font keys, following the pattern defined in CSS models.

```
    #1~font-size/.fontsize         = l__phd_#2_fontsize_tl,
    #1~font-weight/.fontweight     = l__phd_#2_fontweight_tl,
    #1~font-shape/.fontstyle       = l__phd_#2_fontshape_tl,
    #1~font-style/.fontstyle       = l__phd_#2_fontshape_tl,
    #1~font-family/.fontfamily     = l__phd_#2_fontfamily_tl,
    #1~font-face/.fontface         = l__phd_#2_fontface_tl,
```

The ⟨*element*⟩`case` can be used to determine the text case, as lower or upper. This will have to be extended one day for languages that have difficult folding cases.

```
    #1~case/.case                  = l__phd_#2_case_tl,
```

The ⟨*element*⟩`format` is the workhorse of the layout engine. It is similar to templates in other languages. We use a special handler that checks if the format is either in a list or is defined. If it has been registered it can have any value, otherwise it needs to follow the convention, `\format_`⟨*name*⟩:nn.This API is still not very stable and I might change it down the line.

```
    #1~format/.format~in           = l__phd_#2_format_tl,
```

The following, keys deal with colors. Color names are stored in macros, with `.store`. The colorin macro uses `color`⟨*value*⟩. The handler checks that the color exists. If it does not exist it issues an error and maps to the color black. The values `palette` or `auto`, have a special meaning. If a palette is defined the color will map to `the`⟨*element*⟩`color` or to the element parent color. The inheritance still needs to be defined better.

Add missing colornames in color palette.

```
    #1~background-color/.store         = l__phd_#2_background_color_tl,
    #1~frame-color/.store              = l__phd_#2_frame_color_tl,
    #1~color/.colorin                  = l__phd_#2_color_tl,
    #1~shadow/.shadow                  = l__phd_#2_shadow_tl,
```

The following keys deal with setting values for heading elements that use the 'box' model. They are normall defined in a similar way to those defined by **tcolorbox**.

```
    #1~width/.store                    = l__phd_#2_width,
    #1~arc/.store                      = l__phd_#2_arc_tl,
    #1~grow~left/.store                = l__phd_#2_grow_left_dim,
    #1~grow~right/.store               = l__phd_#2_grow_right_dim,
    #1~rounded~corners/.store          = l__phd_#2_rounded_corners_tl,
% Borderlines
    #1~borderline~top/.store           = l__phd_#2_borderline_top_tl,
    #1~borderline~right/.store         = l__phd_#2_borderline_right_tl,
    #1~borderline~bottom/.store        = l__phd_#2_borderline_bottom_tl,
    #1~borderline~left/.store          = l__phd_#2_borderline_left_tl,
```

Borderlines can be colored individually. They can use the palette, auto, cascade values, as previously discussed.

```
      #1~borderline~top-color/.store    = l__phd_#2_borderline_top_color_tl,
      #1~borderline~right-color/.store  = l__phd_#2_borderline_right_color_tl,
      #1~borderline~bottom-color/.store = l__phd_#2_borderline_bottom_color_tl,
      #1~borderline~left-color/.store   = l__phd_#2_borderline_left_color_tl,
```

The ⟨*element*⟩`border-top-width` and other similar elements store the dimension of borders, padding and margins.

```
% Box borders
      #1~border-top-width/.store        = l__phd_#2_border_top_width_dim,
      #1~border-right-width/.store      = l__phd_#2_border_right_width_dim,
      #1~border-bottom-width/.store     = l__phd_#2_border_bottom_width_dim,
      #1~border-left-width/.store       = l__phd_#2_border_left_width_dim,
% Box padding
      #1~padding-top-width/.store       = l__phd_#2_padding_top_width_dim,
      #1~padding-right-width/.store     = l__phd_#2_padding_right_width_dim,
      #1~padding-bottom-width/.store    = l__phd_#2_padding_bottom_width_dim,
      #1~padding-left-width/.store      = l__phd_#2_padding_left_width_dim,
% Box margin
      #1~margin-top-width/.store        = l__phd_#2_margin_top_width_dim,
      #1~margin-right-width/.store      = l__phd_#2_margin_right_width_dim,
      #1~margin-bottom-width/.store     = l__phd_#2_margin_bottom_width_dim,
      #1~margin-left-width/.store       = l__phd_#2_margin_left_width_dim,
    }
  }



\ExplSyntaxOff
```

Chapter, Part and sometimes Sections might need to be typeset on new pages, we create booleans for these,

```
\ExplSyntaxOn
\cs_gset:Npn \make_new_bool:n #1
  {
    \bool_new:c {#1_open_left_bool}
    \bool_gset_false:c {#1_open_left_bool}
    \bool_new:c {#1_open_any_bool}
    \bool_gset_false:c {#1_open_any_bool}
    \bool_new:c {#1part_open_anywhere_bool}
    \bool_gset_false:c {#1_open_anywhere_bool}
    \bool_new:c {#1_open_right_bool}
    \bool_gset_false:c {#1_open_right_bool}
  }

\make_new_bool:n {part}
\make_new_bool:n {chapter}
\make_new_bool:n {section}
\ExplSyntaxOff
```

We create keys for the booleans by defining a function and the calling it for the document elements we allow,

```
\ExplSyntaxOn
\cs_gset:Npn \make_new_opening_keys:n #1
{
 \cxset{
```

```
    #1~opening/.is~choice,
    #1~opening/right/.code      = \bool_gset_true:c {#1_open_right_bool},
    #1~opening/left/.code       = \bool_gset_true:c {#1_open_left_bool},
    #1~opening/any/.code        = \bool_gset_true:c {#1_open_any_bool},
    #1~opening/none/.code       = \bool_gset_true:c {#1_open_anywhere_bool},
    #1~opening/anywhere/.code   = \bool_gset_true:c {#1_open_anywhere_bool},
    #1~opening/ifafter/.code={},
 }
}
\make_new_opening_keys:n {part}
\make_new_opening_keys:n {chapter}
\make_new_opening_keys:n {section}
\ExplSyntaxOff
```

Set the default values for part, chapter and sections.

```
\cxset{part opening = anywhere,
          chapter opening = anywhere,
          section opening = none}
```

`\l_@@_make_new_numbering_keys:n` {⟨*element name*⟩}

Not all heading elements need to access a counter and associated fields, so we provide a function to specifically target these keys.

```
\ExplSyntaxOn
\cs_new:Npn \l__phd_make_new_numbering_keys:n #1
  {
% Numbering Wow!
\cxset{
     #1~number~prefix/.store       = lphd#1numberprefixtl,
     #1~number~suffix/.store       = lphd#1numbersuffixtl,

     #1~number~after/.store        = l__phd_#1_number_after_tl,
     #1~numbering/.is~choice,
     #1~numbering/roman/.code          =
       \cs_gset:cpn {the#1}
         {
           \cs:w lphd#1numberprefixtl \cs_end:
             \@roman\cs:w c@#1\cs_end:\relax
           \cs:w lphd#1numbersuffixtl \cs_end:
         },
     #1~numbering/Roman/.code          =
      \cs_gset:cpn {the#1}
        {
          \cs:w lphd#1numberprefixtl \cs_end:
          \expandafter\@Roman{\cs:w c@#1\cs_end:\relax}
          \cs:w lphd#1numbersuffixtl \cs_end:
        },
    #1~numbering/(roman)/.code          =
    \cs_gset:cpn {the#1}
       {
       \cs:w lphd#1numberprefixtl \cs_end:
         (\@roman\cs:w c@#1\cs_end:\relax)
       \cs:w lphd#1numbersuffixtl \cs_end:
       },
```

```
 #1~numbering/(Roman)/.code          =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
     (\@Roman \cs:w c@#1\cs_end:\relax)
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/arabic/.code           =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
     \@arabic\cs:w c@#1\cs_end: \relax
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/numeric/.code          =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
     \@arabic\cs:w c@#1\cs_end:\relax
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/none/.code             =
 \cs_gset:cpn {the#1} {},

#1~numbering/alpha/.code            =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
     \exp_after:wN \alphalph {\cs:w c@#1\cs_end:\relax}
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/Alpha/.code            =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
       \exp_after:wN \AlphAlph{\cs:w c@#1\cs_end:\relax}
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/words/.code            =
 \cs_gset:cpn {the#1}
   {
    \cs:w lphd#1numberprefixtl \cs_end:
    \words@cx{\@arabic\cs:w c@#1\cs_end:\relax}
    \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/Words/.code            =
 \cs_gset:cpn {the#1}
   {
     \cs:w lphd#1numberprefixtl \cs_end:
     \Words@cx{\@arabic\cs:w c@#1\cs_end:\relax }
     \cs:w lphd#1numbersuffixtl \cs_end:
   },
#1~numbering/WORDS/.code            =
 \cs_gset:cpn {the#1}
```

```
      {
       \cs:w lphd#1numberprefixtl \cs_end:
       \WORDS@cx{\@arabic\cs:w c@#1\cs_end:\relax}
       \cs:w lphd#1numbersuffixtl \cs_end:
      },
    #1~numbering/ordinals/.code            =
    \cs_gset:cpn {the#1}
      {
       \cs:w lphd#1numberprefixtl \cs_end:
       \ordinals@cx{\@arabic\cs:w c@#1\cs_end:\relax}
       \cs:w lphd#1numbersuffixtl \cs_end:
      },
    #1~numbering/ORDINALS/.code            =
    \cs_gset:cpn {the#1}
      {
       \cs:w lphd#1numberprefixtl \cs_end:
       \ORDINALS@cx{\@arabic\cs:w c@#1\cs_end:\relax}
       \cs:w lphd#1numbersuffixtl \cs_end:
      },
   #1~numbering~custom/.code           =
    \cs_gset:cpn {the#1} {##1},
  }
 }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_gset:Npn\phd_setparent:nn #1#2 {%
   \cs_gset:cpn {#1parent} {#2}
}
```

G.5.1 Inheritance, auto and cascade

This can be a list pointing up. also need to thnk about secondary elements For example `\chapterparent` will point to *document* and where keys are set to inherit will be inherited. Other key values set to cascade can go up to top level. This is probably and ideal way to use clists.

```
\phd_setparent:nn {document}{root}
\phd_setparent:nn {part}{document}
\phd_setparent:nn {chapter}{document}
\phd_setparent:nn {section}{chapter}
\phd_setparent:nn {subsection}{section}
\phd_setparent:nn {subsubsection}{subsection}
\phd_setparent:nn {paragraph}{subsubsection}
```

G.5.2 Creating the Keys

We are now ready to start creating the keys for the headings. The headings names are stored in `phd_book_divisions_clist`. We iterate through all the elements to create the keys, first for the major elements and second for the secondary elements. Reminder a major element can be defined with one word and a secondary element with two or more.

```
\clist_map_inline:Nn \__phd_book_divisions_clist
```

```
  {
    \l__phd_make_new_heading_keys:n{#1}{#1}
    \l__phd_make_new_numbering_keys:n{#1}
  }
```

Map through the list again for the secondary elements.

```
\clist_map_inline:Nn \__phd_book_divisions_clist
  {
    \iow_log:n {#1, }
    \l__phd_make_new_heading_keys:n {#1~title}{#1_title}
    \l__phd_make_new_heading_keys:n {#1~title~before}{#1_title_before}
    \l__phd_make_new_heading_keys:n {#1~number}{#1_number}
    \l__phd_make_new_heading_keys:n {#1~number~before}{#1_number_before}
    \l__phd_make_new_heading_keys:n {#1~label}{#1_label}
    \l__phd_make_new_heading_keys:n {#1~label~before}{#1_label_before}
    \l__phd_make_new_heading_keys:n {#1~before}{#1_before}
    \l__phd_make_new_heading_keys:n {#1~after}{#1_after}
  }

\ExplSyntaxOff
```

G.6 Programmatically create key defaults

With such a large list of properties, we create default values programmatically. Then these are overwritten later to more appropriate values. This ensures that all keys are created and that no errors occur.

`\l_@@_set_headings_key_defaults:n` {⟨*element name*⟩}

This function takes one agument and inititalizes all the keys.

```
\ExplSyntaxOn
\cs_new:Npn \l__phd_set_headings_key_defaults:n #1
  {
    \cxset
      {
% name
       % #1~name                    = Chapter,
% numbering
      %  #1~numbering               = arabic,
        #1~align                   = Centering,
% renderer format
        #1~format                  = block,
% fonts
        #1~font-size               = HUGE,
        #1~font-weight             = bold,
        #1~font-family             = sffamily,
        #1~font-shape              = upshape,
        #1~font-style              = upshape,
        #1~case                    = upper,
% colors
        #1~color                   = black,
        #1~background-color        = bgsexy,
        #1~frame-color             = bgsexy,
```

```
% spacing
        #1~indent                  = 0pt,
        #1~beforeskip              = 10pt,
        #1~afterskip               = -3ex,
        #1~afterindent/.onoff      = #1_afterindent,
% box and effects
        #1~arc                     = 0pt,

        #1~grow~left               = 0mm,
        #1~grow~right              = 0mm,
        #1~rounded~corners         = northeast,
        #1~shadow                  = drop~shadow,
% borderlines
        #1~borderline~top           = 1pt,
        #1~borderline~right         = 1pt,
        #1~borderline~bottom        = 1pt,
        #1~borderline~left          = 1pt,
% borderlines color
        #1~borderline~top-color     = black,
        #1~borderline~right-color   = black,
        #1~borderline~bottom-color  = black,
        #1~borderline~left-color    = black,
%
        #1~border-left-width       = 1pt,
        #1~border-right-width      = 1pt,
        #1~border-top-width        = 1pt,
        #1~border-bottom-width     = 1pt,

        #1~padding-left-width      = 1pt,
        #1~padding-right-width     = 1pt,
        #1~padding-top-width       = 1pt,
        #1~padding-bottom-width    = 1pt,
%
      }
  }
```

Next we map all the keys to defaults.

book_divisions_clist

```
\clist_map_inline:Nn \__phd_book_divisions_clist
  { \wlog{set defaults ... #1 ...,}
    \l__phd_set_headings_key_defaults:n {#1}
    \l__phd_set_headings_key_defaults:n {#1~title}
    \l__phd_set_headings_key_defaults:n {#1~title~before}
    \l__phd_set_headings_key_defaults:n {#1~label}
    \l__phd_set_headings_key_defaults:n {#1~label~before}
    \l__phd_set_headings_key_defaults:n {#1~number}
    \l__phd_set_headings_key_defaults:n {#1~number~before}
    \l__phd_set_headings_key_defaults:n {#1~before}
    \l__phd_set_headings_key_defaults:n {#1~after}
  }
\ExplSyntaxOff
```

(*End definition for* `\__phd_book_divisions_clist`.)

```
\ExplSyntaxOn
\cxset
  {
    section~spaceout/.is~choice,
    section~spaceout/soul/.code            = \@sectionspaceouttrue,
    section~spaceout/none/.code            = \@sectionspaceoutfalse,
 }
\ExplSyntaxOff
```

As described earlier boxed headings have numerous elements, each of which can be styled on its own.

Formatters and Associated Functions

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

Formatters are functions that produce a layout and apply the properties to a number of elements. There are is a formatter for each format available as a property to a heading. The most comprehensive is the css block formatter that can size individual boxes and also float them as required. Special formatters can be added to the package as extensions.

H.1 Helper functions

The helper functions transfer properties form keys to tikz or tcolorbox boxes or to plain macros for simpler formatters. They are just convenient functions to keep the code for the formatters, shorter.

`\set_font_aux:n` {⟨*element name*⟩}

This helper function sets the font properties.

```
\ExplSyntaxOn
\cs_gset:Npn \set_font_aux:n #1
 {
   %\cs_if_exist_use:c { #1_fontface_tl}
   \cs_if_exist_use:c { #1_fontfamily_tl }
   \cs_if_exist_use:c { #1_fontweight_tl }
   \cs_if_exist_use:c { #1_fontshape_tl  }
   \cs_if_exist_use:c { #1_fontsize_tl   }

 }
\cs_new:Npn \set_borders_aux:nn #1 #2
 {
   \dim_if_exist:cTF{l__phd_#1_border_#2_width_dim } {8pt}{2pt}
  }
```

`\set_color_aux:nn` {⟨*element name*⟩} {⟨⟩}

```
\cs_new:Npn \set_color_aux:nn #1 #2
 {
   \cs:w l__phd_#1_#2_color_tl \cs_end:
 }
\ExplSyntaxOff
```

`\set_borderline_aux:nn` {⟨⟩} {⟨⟩}

Sets the borders of a tcolorbox or other box. For tcolorbox, there is an important difference between borderlines and rules. Borderlines are a bit problematic in a declarative interface. Firstly they take four arguments. Following the shadows interface we offer predetermined, styles as well as allow maximum flexibility through styles for the user.
To draw them they require offset, thikness, color and line type. The offset determines if they drawn away or inwards the original path.

```
\ExplSyntaxOn
\cs_new:Npn \set_borderline_aux:nn #1 #2
{
```

```
  \tcbset{borderline~style/.style =
  {
    borderline~west = {\cs:w l__phd_#1_borderline_left_tl \cs_end:}
                      {-\set_borders_aux:nn {#1} {left}-0pt }
                      {
                       \cs:w l__phd_#1_borderline_left_color_tl \cs_end: ,
                        solid,
                      },
    borderline~east = {\cs:w l__phd_#1_borderline_right_tl \cs_end:}
                      {-\set_borders_aux:nn {#1} {right}-0pt }
                      {
                        \cs:w l__phd_#1_borderline_right_color_tl \cs_end:,
                        solid
                      },
    borderline~north = {\cs:w l__phd_#1_borderline_top_tl \cs_end:}
                      {-\set_borders_aux:nn {#1} {top}-0pt }
                      { \cs:w l__phd_#1_borderline_top_color_tl \cs_end:,
                       dashed},
    borderline~south = {\cs:w l__phd_#1_borderline_bottom_tl \cs_end:}
                      {-\set_borders_aux:nn {#1} {bottom}-0pt }
                      {
                        \cs:w l__phd_#1_borderline_bottom_color_tl \cs_end:,
                        dashed
                      },
   }
 }
}
\ExplSyntaxOff
```

`\format_inmargin`

This is a simple formatter that places the heading in the margin. Useful for old style books.

```
\ExplSyntaxOn
\cs_set:Npn \format_inmargin:nnn #1#2#3
  {
    \tcbdocmarginnote
    {

      \hbox{Section~\@svsec}
      #3
    }
  }
\ExplSyntaxOff
```

H.2 Hanged formats

Most books will have sections as hanged paragraphs. They can be indented or outdented into the margins and can consist of a single line or more. We provide a genereric formatter based on the one in LaTeX first.

H.2 Test A section

`\format_hang:nn` {⟨*element name*⟩} {⟨*unused*⟩} {⟨*title text*⟩}

This formatter renders the classic LaTeX hanged paragraph. As all the properties are available in the keys, we do not really need to concern ourselves with passing values from the main function to here.

```
\ExplSyntaxOn
\cs_set:Npn \format_hang:nn #1#2
  {\par\leavevmode
    \phd_hang_from:nn
        {
         \set_font_aux:n {l__phd_#1}
         \cs_if_exist_use:c {the_#1_title_fontface_tl}
         \cs:w
          l__phd_#1_number_color_tl
          \cs_end:
         {\hskip 0pt %indent
           \cs:w the#1 \cs_end:
           \hskip1em
         }
        }
        {
          \set_font_aux:n {l__phd_#1}
          \cs_if_exist_use:c {the_section_title_fontface_tl}
          \cs:w
          l__phd_#1_color_tl
          \cs_end:
          #2

        }
   { \interlinepenalty\@M
    \par } %Check par here
```

In classic LaTeX end of section action depends on the sign of the afterskip. A negative sign is used to indicate an inline paragraph and a positive a hanged section. All the post action being handled by `\@xsect`.

As this is not very intuitive, we move the xsect code below for the hang paragraphs, with the intention to simplify the inline code in the inline renderer later.

```
    \par \nobreak
    %\skip_vertical:c {l__phd_#1_afterskip_tl}
    \vskip10pt
    \phd_after_heading:
    \tex_ignorespaces:D
  }
\ExplSyntaxOff
```

`\format_hang_inmargin` {⟨*element name*⟩} {⟨*text*⟩}

This is a variation to the hanged format, where the heading number is in the margin.

```
\ExplSyntaxOn
\cs_set:Npn \format_hang_inmargin:nn #1#2
```

```
  {
    \bgroup
    \parindent0pt
    \setbox\@tempboxb\hbox
     {
       \set_font_aux:n {l__phd_#1}
       \expandafter\@arabic\cs:w c@#1 \cs_end:
     }
     \makebox[0pt]{
          \makebox[-\wd\@tempboxb][r]{%
         \set_font_aux:n {l__phd_#1}
         {
           \tcbox[nobeforeafter,box~align=center,
                  colback=bgsexy]
                  {
                    \cs:w the#1 \cs_end:
                  }
         }
        }
        }
     \begin{tcolorbox}[enhanced,
                       %title empty,
                        size=minimal,
                        nobeforeafter,box~align=center,
                        title,title code={
\path[draw=yellow,solid,decorate,line width=2mm,
decoration={coil,aspect=0,segment length=10.1mm}]
([xshift=1mm]title.west) -- ([xshift=-1mm]title.east);}
                        ]
          \set_font_aux:n {l__phd_#1}
          \cs:w
          l__phd_#1_color_tl
          \cs_end:
          #2
    \end{tcolorbox}
    \egroup
    \interlinepenalty\@M
    \tex_par:D
    \par \nobreak
    \skip_vertical:c {l__phd_#1_afterskip_tl}
    \phd_after_heading:
  }
\ExplSyntaxOff
```

H.3 Format simple ruled

A great number of variations to heads are simple rules and text. For these we can increase rendering speed by direct coding.

```
\ExplSyntaxOn
\def\thickhrulefill{\leavevmode \leaders \hrule height 1ex \hfill \kern \z@}
\cs_set:Npn \zoonek #1 #2{%
  \vspace*{10\p@}%
```

```
  {\parindent \z@ \centering \reset@font

        {\Huge \scshape \csname the#1\endcsname}
        \par\nobreak
        \vspace*{10\p@}%
        \interlinepenalty\@M
        \thickhrulefill
        \par\nobreak
        \vspace*{2\p@}%
        % title
        {\Huge \bfseries #2\par\nobreak}
        \par\nobreak
        \vspace*{2\p@}%
        \thickhrulefill
    %\vskip 40\p@
    \vskip 100\p@
  }}

\cs_set:Npn \zoonekii #1 #2 {
 {\parindent \z@ \raggedright
    \reset@font
    \hrule
    \vspace*{10\p@}%
    \par
    \Large \scshape \@chapapp{}\hspace{0.5em}
    \Huge\bfseries \csname the#1\endcsname
    \par\nobreak
    \vspace*{10\p@}%
    \hrule
    \par
    \vspace*{1\p@}%
    \hrule
    %\vskip 40\p@
    \vspace*{20\p@}
    \Huge \bfseries #2\par\nobreak
    \vskip 70\p@
 }
}

\cs_set:Npn \zoonekiii #1 #2 {
 \vspace*{10\p@}%
  {\parindent \z@ \raggedleft \reset@font
            \scshape \gettext{danish}{#1}~\csname the#1\endcsname
        \par\nobreak
        \interlinepenalty\@M
    \Huge \bfseries #2\par\nobreak
    %\vspace*{1\p@}%
    \hrulefill
    \par\nobreak
    \vskip 100\p@
  }


}
```

```

\cs_set:Npn \zoonekiv #1 #2
{
{
\reset@font
  \parindent \z@
  \vspace*{10\p@}%
  \hbox{%
    \vbox{\hsize=2cm
      \begin{tabular}{c}
        \scshape \strut \gettext{greek}{#1}\\ %\@chapapp{} \\
        \fbox{%
          \vrule depth 10em width 0pt%
          \vrule height 0pt depth 0pt width 1ex%
          {\LARGE \bfseries \strut \csname the#1\endcsname}%
          \vrule height 0pt depth 0pt width 1ex%
          }
      \end{tabular}%
      }%
    \vbox{%
      \advance\hsize by -2cm
      \hrule\par
      \vskip 6pt%
      \hspace{1em}%
      \Huge \bfseries #2
      }%
    }%
  \vskip 100\p@
}
}
```

The next formatter will probably benefit if changed to tikz. Left it here as it demonstrates ingenuity in using a rule and then overlaying it with the hading number.

It can also benefit with the use fo color. It will have to be mirrored for RTL text.

```
\def\LigneVerticale{\vrule height 5cm depth 2cm\hspace{0.1cm}\relax}
\def\LignesVerticales{%
  \let\LV\LigneVerticale\LV\LV\LV\LV\LV\LV\LV\LV\LV\LV}

\def\GrosCarreAvecUnChiffre#1{%
  \rlap{\vrule height 0.8cm width 1cm depth 0.2cm}%
  \rlap{\hbox to 1cm{\hss\mbox{\color{white}#1}\hss}}%
  \vrule height 0pt width 1cm depth 0pt}

\cs_set:Npn \zoonekv #1 #2
{
 \hbox~to~\linewidth{\hss\textdir TLT
    \huge
    \LignesVerticales
    \hspace{-0.5cm}%
    \GrosCarreAvecUnChiffre{\thechapter}
    \hspace{0.2cm}
    \hbox{\parbox{10cm}{#2}}%
```

```
  }\par\vskip 2cm
}
\definecolor{red1}{rgb}{1,0.9,0.9} \definecolor{blue1}{rgb}{0.9,0.9,1} \definecolor{gree
\definecolor{yellow1}{rgb}{1,1,0.8} \definecolor{yellow2}{rgb}{1,1,0.8}
\definecolor{color11}{rgb}{1,0.9,0.6}
\def\lecture#1#2{\reset@font \fboxsep2mm\vspace{2mm} \begin{center} \fcolorbox{green1}{g

\ExplSyntaxOff
```

E-320: Math, a Historical Perspective **O. Knill, 2010-2018**

Lecture:H Lecture: How to Use Formatters

H.4 Format block

`\format_block:nnnn` ⟨*element name*⟩ ⟨⟩ {⟨⟩} {⟨⟩}

This function is the main function for block and fancy headings. It uses the package **tcolorbox** in order to provide as much fleixibility as possible.

We create a boolean that determines if the element label and its number are to betypeset inline or not.

```
\ExplSyntaxOn
\bool_new:N \combo_if_bool \bool_gset_true:N \combo_if_bool
```

We are now ready to define the function.

```
\cs_set:Npn \label_number_box #1 {
 \bool_if:NTF \combo_if_bool
     {
         \fboxsep=2pt
         \fboxrule=1pt
         %\phd_float_box:nnn {chapter_number} {}{
                \tcbox[size=minimal,
                nobeforeafter,
                before = \vskip1sp,
                colback=white]
                {
                   %\cs:w l__phd_#1_fontsize_tl\cs_end:
                   \tcbox[size=minimal,
                          nobeforeafter,
                          colback= \cs:w l__phd_chapter_label_background_color_tl \cs_en
                          valign=top,
                          ]{
                     \bgroup
                     \cs_if_exist_use:c { the_#1_fontface_tl}
                     \cs_if_exist_use:c {l__phd_#1_label_fontfamily_tl}
                     \cs_if_exist_use:c {l__phd_ #1_label_fontweight_tl}
                     \cs_if_exist_use:c {l__phd_#1_label_fontshape_tl}
                     \cs_if_exist_use:c {l__phd_#1_label_fontsize_tl}

                     %\l__phd_chapter_label_fontfamily_tl
                     %\l__phd_chapter_label_fontshape_tl\space
                     \cs:w l__phd_#1_label_color_tl\cs_end:

                     \cs:w #1name\cs_end:\space

                   \egroup}
```

```

                    \tcbox[size=minimal,
                           nobeforeafter, top=0pt,valign=top,
                           colback=\cs:w
                                l__phd_chapter_number_background_color_tl
                           \cs_end:,
                           ]{
                      \bgroup

                      \cs_if_exist_use:c { the_#1_number_fontface_tl    }
                      \cs_if_exist_use:c {l__phd_#1_number_fontfamily_tl }
                      \cs_if_exist_use:c {l__phd_#1_number_fontweight_tl }
                      \cs_if_exist_use:c {l__phd_#1_number_fontshape_tl  }
                      \cs_if_exist_use:c {l__phd_#1_number_fontsize_tl   }
                      \cs:w l__phd_#1_number_color_tl\cs_end:
```

We have to introduce a phantom here to have a proper strut. Intorduce after the font sizing and use the chapter name?

```
                      \vphantom{\chaptername}
                      \cs:w the#1\cs_end:\scan_stop:
                      \egroup
                     }
               }
            %}
    }
    {
     \phd_float_box:nnn {#1}{}
       {
         \tcbox{\@svsec}

       }
    }
}



\cs_set:Npn \format_block:nnnn #1#2#3#4
  {
    \bgroup
    %\leftskip-1.25in
    \set_borderline_aux:nn {#1}{}
    \noindent\begin{tcolorbox}[size=minimal,
      width       = \textwidth-4pt, %\pdfpagewidth-4cm,
      %arc         = 10mm,
      %auto~outer~arc,
      \cs_if_exist_use:cTF {l__phd_#1_shadow_tl}{}{},
        colback           = white,  %bgsexy!5,%\set_color_aux:nn {#1}{background},FIXME
        colframe          = white, %\set_color_aux:nn {#1}{frame},
        toprule           = \set_borders_aux:nn {#1} {top},
        rightrule         = \set_borders_aux:nn {#1} {right},
        bottomrule        = \set_borders_aux:nn {#1} {bottom},
        leftrule          = 0pt,%\set_borders_aux:nn {#1} {left},
        left~skip =0pt,
        left=0pt,
```

```
        boxsep=0pt,
      %  borderline        = {5pt}{5pt}{blue,double,dashed,line~join=bevel},
     %  borderline~north = {2pt}{-3pt}{dotted},
     %  borderline~south = {2pt}{-3pt}{dotted},
     %  borderline~east  = {2pt}{-3pt}{dotted},
     %  borderline~west  = {2pt}{-3pt}{dotted},
      % borderline~style,
       % no~borderline,
       % show~bounding~box,
        halign = left,
        valign = top,
        %rounded~corners=all,
        %drop~lifted~shadow=black,
     ]

    \label_number_box {#1}
%
% above title box {empty}
   \phd_float_box:nnn {#1_title_before}{}{}
% title float box

  \phd_float_box:nnn {#1_title}{}{#4}
   \par
   \vskip10pt %TODO
   \end{tcolorbox}
   \vskip10pt
 \egroup
 \par\nobreak\nointerlineskip
}
```

\phd_float_box:nnn {⟨*identifier*⟩} {⟨⟩} {⟨⟩}

#1 The section label

#2 The secondary identifier i.e, title

#3 The text of the box

This box contains an outer and an inner box, permitting the second box to float freely into the first box. The widths are constrained based on user inputs or automatic calculations.

```
\cs_set:Npn \phd_float_box:nnn #1 #2 #3
{\set_borderline_aux:nn {#1}{}

  \begin{tcolorbox}
    [
      size=minimal,
      width = \linewidth,
      no~shadow,
      colback  = \cs:w l__phd_#1_background_color_tl \cs_end:,
      colframe = \cs:w l__phd_#1_frame_color_tl \cs_end:,
      toprule  = 0pt,
      borderline~style,
      boxsep=0pt,
```

```
      left~skip=0pt,
      left=0pt,
      right=0pt,
      right~skip=0pt,
     ]

    \set_font_aux:n {l__phd_#1}
    \cs:w l__phd_#1_align_tl \cs_end:
    %in between box or title or author block
    \begin{tcolorbox}
      [
        %#1~outer,
        size=minimal,
        no~shadow,
        colback = white,%\cs:w l__phd_#1_background_color_tl \cs_end:,
        colframe= white,%\cs:w l__phd_#1_frame_color_tl \cs_end:,,
        boxsep=0pt,
        top=0pt,
        bottom=0pt,
        left=0pt,
        right=2pt,
        %drop~shadow,
        %width=\linewidth,
        borderline~style,
       % halign = ,
       ]
       {   \language-1\relax
       %\RaggedRight
      \set_font_aux:n {l__phd_#1_tl}
      \cs:w  l__phd_#1_color_tl  \cs_end:
      \cs:w  l__phd_#1_align_tl  \cs_end:
       {\textsl{\textls{#3}}}\relax %
       \vskip10pt
       }
  \end{tcolorbox}
    \par
  \end{tcolorbox}
}

\ExplSyntaxOff
```

H.4.1 Display format

The display format typesets a heading in a similar fashion to traditional chapters.

`\format_display:nn`{⟨*section name*⟩} {⟨*skip after number*⟩} {⟨⟩} Displays a section similar to Chapters

#1 Section name
#2 indent

#3 format para
#4 Title text
svsec number

```
\ExplSyntaxOn
\cs_set:Npn \format_display:nnnn #1 #2 #3 #4
{
  %\cxset{#1~title~margin-top-width=30pt}
  \format_block:nnnn {#1}{#2}{#3}{#4}
}
 \ExplSyntaxOff
```

#1 name #2 indent #3 title

```
\ExplSyntaxOn
\cs_set:Npn \format_inline:nnn #1 #2 #3
  {
   {\bfseries\normalfont
    \theparagraph #3}
   }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_new:Npn \format_part_traditional:nn #1 #2 #3
  {\par\leavevmode
   \group_begin:
      \centering
     \interlinepenalty \@M
     \set_font_aux:n {l__phd_#1}
     \cs:w l__phd_#1_number_tl\cs_end:
     \set_font_aux:n {l__phd_#1_number}
     \ifnum \c@secnumdepth >-2\relax
       \csname #1name\endcsname\nobreakspace\csname the#1\endcsname
       \par
       \vskip 20\p@
     \fi
     \group_begin:
       %\set_color:nn {#1}{color}
       \set_font_aux:n {l__phd_#1 }
          #3
     \group_end:
     \tex_par:D
     \vskip20pt
   \group_end:

  }
\ExplSyntaxOff
```

```
\ExplSyntaxOn
\cs_new:Npn \format_traditional:nn #1 #2 #3
  {\par\leavevmode
   \group_begin:
    \centering
     \interlinepenalty \@M
     \set_font_aux:n {l__phd_#1}
     \cs:w l__phd_#1_number_tl\cs_end:

```

```
      \ifnum \c@secnumdepth >-2\relax
        \set_font_aux:n {l__phd_#1_number}

        \csname #1name\endcsname\nobreakspace\csname the#1\endcsname
        {
          \cxset{chapter~label~font-family=\ovidius}
          \begin{tikzpicture}
          \node (a) at (0,0) {\expandafter\csname#1name\endcsname};
          \node (b) at (0,15pt) {\tiny \set_font_aux:n{l__phd_chapter_label} test};
         \end{tikzpicture}
        }
        \par
        \vskip 20\p@
      \fi
      \group_begin:
        %\set_color:nn {#1}{color}
        \set_font_aux:n {l__phd_#1 }
           #3
      \group_end:
      \par
      \vskip20pt
    \group_end:

  }
\ExplSyntaxOff
```

I

Layout Engine Code

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

The Layout Engine renders a heading, made out of elements. It uses some of the techniques of LaTeX, primarily for me to understand the methodology used in the old kernel.

The function is declared using **xparse** to simplify the switching between the various options of the headings commands.

`\start_section:nnnnnnnnn` {⟨*nine parameters*⟩}

The command takes up nine parameters

`#1` name e.g, subsection
`#2` level: a number, denoting depth of a section - e.g., chapter=0, section=1, etc.
`#3` indent: Indentation of heading from left margin.
`#4` beforeskip: Absolute value = skip to leave above the heading. If negative, then paragraph indent of the text following the heading is supressed.
`#5` afterskip:
`#6` styling command
`#7` star or absent
`#8` optional argument
`#9` section title

```
\ExplSyntaxOn
\DeclareDocumentCommand \start_section:nnnnnnnnn {m m m m m m s o m}
  {
    \if@noskipsec \leavevmode \fi
    \par
```

The old LaTeX code used to check if the skip was negative (i.e., for inline Paragraph headings to decide if it will indent or not afterwards. We have a boolean for every heading, allowing us for example for subsubsections to indent, so this code is modified slightly.

```
%    check for before skip
    \l_tmpa_skip #4\relax
    \@afterindenttrue
    \if_dim:w \l_tmpa_skip <\z@
% make it positive
      \skip_gset:Nn\l_tmpa_skip {-\l_tmpa_skip}
      \@afterindentfalse

    \fi:

    \bool_if:cTF {#1_afterindent}{
       \@afterindentfalse
    }{\@afterindenttrue}
%
    \if@nobreak
      \everypar{\tikzi[start\\sect] \bool_if:cTF {#1_afterindent}{\tikzi[TRUE]}{\tikzi[
    \else
```

```
      \addpenalty \@secpenalty
      \addvspace\l_tmpa_skip
    \fi
%
% redirect depending on star or option
%
    \IfBooleanTF {#7}
    % send to star section
      {
      % This is only for LaTeX Compatibility
      % name is not send which is problematic as we cannot
      % pick-up the properties
      %
         \gdef\currentsectionname{#1}
         %\@ssect {#3} {#4} {#5} {#6} {#9}
         \@sect:  {#1} {#2} {#3} {#4} {#5} {#6} {*} [{#8}] {#9}

      }
      {
       \@sect: {#1} {#2} {#3} {#4} {#5} {#6} {*} [{#8}] {#9}
      %{\@sect:  {#1} {#2} {#3} {#4} {#5} {#6} {#7} [{#8}] {#9} }
      }
  }
```

If we are running in compatibility mode we totally ignore the function and revert back to normal LaTeX.

```
\@ltxcompatfalse
\if@ltxcompat
  \else
  \cs_gset_eq:NN \@startsection \start_section:nnnnnnnnn
\fi

\ExplSyntaxOff
```

\@sect: {⟨*#1*⟩}⟨...⟩{⟨*#9*⟩}

The auxiliary function to start the rendering process.

The function still takes nine parameters as we want to preserve state.

- #1 = name e.g, subsection
- #2 = level: a number, denoting depth of a section - e.g., chapter=0, section=1, etc.
- #3 = indent: Indentation of heading from left margin.
- #4 = beforeskip: Absolute value = skip to leave above the heading. If negative, then paragraph indent of the text following the heading is supressed.
- #5 = afterskip:
- #6 = styling command
- #7 = star or absent
- #8 = optional argument

#9 = section title

```
\ExplSyntaxOn
\cs_set:Npn \@sect: #1 #2 #3 #4 #5 #6 #7 [#8] #9
  {
```

We check if the section number depth allows for the heading to be added to the toc, if not we let it to empty.

```
    \int_compare:nTF {#2>\c@secnumdepth}
      {
         \let\@svsec\@empty
      }
      {
        \refstepcounter{#1}
% Since |@seccnformat| might end with an improper |\hskip| which is scannning
% forward for plus or minus we we end the definition with a relax.
        \protected@edef\@svsec
          {
            \@seccntformat{#1}\relax
          }
        % add short title or long title
        \IfValueTF{#8}
          {
             \cs:w #1mark\cs_end: {#9}
%
             \addcontentsline{toc}{#1}{
                \protect\numberline{\csname the#1\endcsname}{#9}}
          }
          {
             \cs:w #1mark\cs_end: {#9}
              \addcontentsline{toc}{#1}{
                \protect\numberline{\csname the#1\endcsname}{#9}}
          }
      }
  % \@tempskipa #5\relax
% save the title.
   \gdef\g__phd_saved_heading_tl{#9}
```

When we test if the command has a star the star gets absorbed and thrown away we want to retain it for the router, so it can call special versions of the commands with a star or non-star version of the commands.

```
\IfBooleanTF{#7}
    {\phd_format_router:nn *{#1}{#9}  }
    {\phd_format_router:nn {#1}{#9}   }
}
\ExplSyntaxOff
```

I.1 Router

As explained earlier the router will direct the rendering to special macros based on the document division formatter. Each formatter is governed by a minimum of two macros. The first one directs to the main function which then will call the non-star command or the star function auxiliary. Since in most cases the star version will just not print the heading number, using a boolean will save one macro. Following this method generalized versions can be produced using xparse easily and hence increasing the extensibility of the package.

`\phd_format_router:nn`⟨*⟩ {⟨*section name*⟩} {⟨*title*⟩}

The format router selects the format to be used to render the heading. It takes the text as its parameter. The rendering properties are picked up by the template.

```
\ExplSyntaxOn
\DeclareDocumentCommand \phd_format_router:nn {s m m}
  {
    \str_case_x:nnTF {\cs:w l__phd_#2_format_tl \cs_end:}
      {
          { display      } {\format_display:nnnn { #2 } {  } {} { #3     }
                              \xsect:n {3.5ex}{}   }
          { block        } { \format_block:nnnn  {#2 } {  } {} { {#3}    }
                              \xsect:n {3.5ex}{#3}
                            }
          { plain        } {
                              \format_hang:nn    {#2}  {#3}
                              \xsect:n {3.5ex}{#3}
                            }
          {hang} {
                              \IfBooleanTF{#1}
                                { \format_hang:nn * {#2} {#3} }
                                 {\format_hang:nn {#2} {#3}}
                                 \xsect:n {5pt}{section}
                             }

          { hanginmargin } {\format_hang_inmargin:nn {#2} {#3}           }
          { leftmargin   } {\format_hang_inmargin:nn {#2} {#3}           }
          { inline       } {  \xsect:n {-1ex}{#2}                        }
          { inmargin     } {\format_inmargin:nnn {#2} {} {#3}            }
          { traditional  } {\format_part_traditional:nn {#2}{}{#3}
                                          \xsect:n {5pt}{section}         }
          { paperback }  {\format_traditional:nn {#2}{}{#3}
                            \xsect:n {5pt}{section}                       }
```

For special templates, it is the responsibility of the formatter to deal with star and unstar versions of commands. It is also its responsibilty to provide the toc and the marks, changing totally the universal nature of the sectioning commands. This means that we have to add the star back if necessary.

```
          { fashion  }   {
                           \IfBooleanTF{#1}
                             {\fashion*{#2}{#3}}
                             {\fashion{#2}{#3}}

                         }
      }
      {       } % true code
```

We test for formatters provided as extensions. If they have been defined by the user, they can either exist in a file or have been entered in the preamble. We first test for the preamble. If they exist we load them.

```

      {
        \cs_if_exist:cTF { l__phd_#2_format_tl }
          {
            \expandafter\cs:w
               \cs:w l__phd_#2_format_tl
               \cs_end:
               \cs_end:
               %\stewart
               { #2 } { #3 }
          }
          {
          Test empty:\csname l__phd_#2_format_tl\endcsname\\
          \format_hang:nn {#2}{#3}

          }
      }
  }

\ExplSyntaxOff
```

\@ssect ⟨*indent*⟩ ⟨*beforeskip*⟩ ⟨*afterskip*⟩ ⟨*styling commands*⟩ ⟨*arg1*⟩

This is the star verson of the command. What it means is that we want a heading with no numbers and not in the toc. Also it does not add it as a mark! This is very limiting as originally programmed in the kernel; probably the thinking was to use it to create same style headings, that one would use for purposes other than sectioning. In reality many books have unnumbered sections and one might want them to go on the headings. We modify it to be able to do both based on a settings command. So to summarize star section means unumbered. Will use choices as to what must be done with it.

#1 indent
#2 beforeskip
#3 afterskip
#4 styling command
#5 arg1 follows

```
\ExplSyntaxOn
%
\cs_set:Npn \@ssect #1 #2 #3 #4 #5 {
  \@tempskipa #3\relax
  \ifdim \@tempskipa>\z@
  \begingroup
    #4
    {
      \phd_hang_from:nn{\hskip #1}%
      \interlinepenalty \@M

      % do color
%      \newfontfamily\aegean{aegean}
%      \cxset{section~title~color=thesectiontitlecolor,
%             section~title~font-shape=itshape,
%             section~title~font-family=sfamily}

      \cs:w l__phd_\currentsectionname _title_color_tl\cs_end:
      %\pgfkeysvalueof{/phd/section~title~color}
      \set_font_aux:n{l__phd_\currentsectionname _title}
     % \pgfkeysvalueof{/phd/section~title~font-family}
      %\MakeUppercase{#5}
      #5
      \@@par
   }%
  \endgroup
  \else

  \def\@svsechd{#4{\hskip #1\relax #5}}%
  \fi
% |\xsect:n{afterskip}| then sets the afteskipping as well as the afterindent.
  \cxset{section~afterindent = true,
             subsection~afterindent = true,
             subsubsection~afterindent = true}
  \xsect:n{#3}{#1} %ONLY NEEDED FOR HANG PARA
}
%



\ExplSyntaxOff

```

\@xsect:n{⟨*afterskip*⟩}

This command sets handles indentation after a sectioning command. It also handles the printing of the title for inline sections (it is saved as `\@svsechd` earlier. It is common for both the star and unstarred versions of `\section`.

Note that `\@noskipsec` is a switch set true by a sectioning command when it is creating an in-text heading with `\everypar`. `#2` is experimental will fail with people that have redefined the routines

```
\ExplSyntaxOn
\cs_set:Npn \xsect:n #1 #2
  {
  \l_tmpa_skip #1\relax
  \if_dim:w \l_tmpa_skip>-1pt %WATCH better boolean
    \par \nobreak
    \vskip\l_tmpa_skip
    \phd_after_heading:

   % negative dimension
   \else:
    \@nobreakfalse
    \global\@noskipsectrue
    \tex_everypar:D
      {\if@noskipsec
          \global\@noskipsecfalse%resets switch
          {\setbox\z@\lastbox}
          \tex_clubpenalty:D\@M
          \group_begin:
            \parindent0pt
                   \tcbox[size=minimal,
                    nobeforeafter,
                    colback=white,
                    box~align=base]{\bfseries \g__phd_saved_heading_tl }%}
          \group_end:
          \tex_unskip:D
          \l_tmpa_skip #1\relax
          \hskip -\l_tmpa_skip
        \else
          \tex_clubpenalty:D \@clubpenalty
          \tex_everypar:D {\tikzi[every\bool_if:cTF {#2_afterindent}{#2~TRUE}{#2~FALSE/
        \fi
      }
  \fi:
  \tex_ignorespaces:D
  }

\cs_set:Npn \after_block:n #1
 {
   \par \nobreak
    \vskip\l_tmpa_skip
    \phd_after_heading:
 }
\ExplSyntaxOff
```

`\phd_after_heading:` ⟨*void*⟩

This is an explorified version of the LaTeX `\@afterheading` control sequence. It has an interesting purpose. It is called after block headings to remove the indentation from the first line of the next paragraph.

The nobreak hackery is used to ensure that there is no page break after a heading (it will look very odd). We add some hooks for visual debugging. Note

that **\nobreak**, **\break** and **\allowbreak** are just macros holding penalty values.

Also note the definitions of the kernel.

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

```
\ExplSyntaxOn
\cs_gset:Npn \phd_after_heading:
  {
    \@nobreaktrue
    \tex_everypar:D
      {
        \if@nobreak
          \@nobreakfalse
          \clubpenalty \@M %1000 no orphans from this point onwards
          \if@afterindent
          \else
```

If afterindent is true, the indentation is removed. this will have to be changed per heading level.

```
            {\setbox\z@\lastbox}%
            \tikzi[afterheading\\
                   clubpenalty\\
                 = \the\clubpenalty]
        \fi
      \else
      \clubpenalty \@clubpenalty %150
      \tex_everypar:D
        {
          \tikzi[everypar in after heading cleared\\
                 clubpenalty\\ \the\clubpenalty
                ]
        }
      \fi
    }
  }
\ExplSyntaxOff
```

When LaTeX is typesetting the section number it calls **\@seccntformat** to use it when typsetting a section heading number. This is common for all the subsectioning commands. We modify it based on code from **sectsty**

in order to generalize it.

We first check if ⟨*section*⟩**@cntformat** is defined and then we redirect to specific section level command.

\@seccntformat{⟨*section name*⟩}

This is a LaTeX kernel factory command that produces **thesection** etc. In the kernel it only takes a generic value, where we have

`section_number_after_tl`. We modify to enable adjustable values for all sectioning commands.

```
\ExplSyntaxOn
 \cs_gset:Npn \@seccntformat #1
 {
  \@ifundefined{#1@cntformat}%
  {\csname the#1\endcsname\section_number_after_tl}% default
  {\csname #1_cntformat\endcsname}% individual control
 }
```

`\section_number_after_tl` ⟨*void*⟩

This function and its siblings are auxiliary functions.

```
\tl_set:Nn  \section_number_after_tl{\quad}%default value only space
\tl_set:Nn  \subsection_number_after_tl{\quad}%default value only space
\tl_set:Nn  \subsubsection_number_after_tl{\quad}%default value only space
\tl_set:Nn  \l__phd_paragraph_number_after_tl{\quad}%default value only space
\tl_set:Nn  \subparagraph_number_after_tl{\quad}%default value only space
%
\cs_set:Npn \section_cntformat{\thesection\section_number_after_tl}
\cs_set:Npn \subsection_cntformat{\thesubsection\subsection_number_after_tl}
\cs_set:Npn \subsubsection_cntformat{\thesubsubsection\subsubsection_number_after_tl}
\cs_set:Npn \paragraph_cntformat {\theparagraph\l__phd_paragraph_number_after_tl }
\cs_set:Npn \subparagraph_cntformat {\thesubparagraph\subparagraph_number_after_tl }
\ExplSyntaxOff
```

`\part` ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

The standard LaTeX part is renewed next. We avoid all the complications of the kernel with different longish sections for chapters and parts.

```
\ExplSyntaxOn
\cs_gset:Npn \emptypagecontents{This~is~an~empty~page.}
\ExplSyntaxOff


\NewDocumentCommand\EmptyPageContents{ m }{
\global\long\def\emptypagecontents {#1}
}

\EmptyPageContents{%
  \vspace*{\fill}%
  \begin{center}
  \LARGE \lorem
  \end{center}
  \vspace{\fill}%
  \thispagestyle{empty}
}
```

`cleardoublepage`

```
\long\def\cleardoublepage{\clearpage\if@twoside\ifodd\c@page\else
  \hbox{}
```

```
    \emptypagecontents\tikzi[In Clear Double Page, empty page]
  \newpage
  \if@twocolumn\hbox{}\newpage\fi\fi\fi}
```

(*End definition for* `cleardoublepage`.)

\part

```
\ExplSyntaxOn
 \renewcommand\part {%
   \newpage\null
   \thispagestyle{empty}
    \start_section:nnnnnnnnn{part}%
      {-1}  %level check this conflicts with source2e
      {\l__phd_part_indent_tl} %indent#2

      {\l__phd_part_beforeskip_tl}%before skip#3

      {\l__phd_part_afterskip_tl}% after skip#4

      {
%        \setfont@cx
%        {\l__phd_chapter_fontweight_tl}%
%        {\l__phd_chapter_fontfamily_tl}
%        {\l__phd_chapter_fontsize_tl}
%        {\l__phd_chapter_fontshape_tl}%
          %\expandafter\setfontparam@cx\l__phd_chapter_align_tl;%
          \l__phd_part_color_tl %5
      }
 }
\ExplSyntaxOff
```

(*End definition for* `\part`.)

```
\ExplSyntaxOn
\cs_gset:Npn \phd_set_opening #1 {
    \bool_if:cT {chapter_open_left_bool}  {
       \cleartoevenpage\null
       \global\@topnum\z@~
       \tikzi[open~left]
       \bool_gset_false:c {#1_open_right_bool}
       \bool_gset_false:c {#1_open_any_bool}
       \bool_gset_false:c {#1_open_anywhere_bool}
       }
    \bool_if:cT {#1_open_right_bool}
      {
        \cleardoublepage\null
        \global\@topnum\z@~
        \tikzi[open~right]
        \bool_gset_false:c {#1_open_left_bool}
        \bool_gset_false:c {#1_open_any_bool}
        \bool_gset_false:c {#1_open_anywhere_bool}
      }
    \bool_if:cT {#1_open_any_bool}
      {
        \clearpage\null
```

```
        \global\@topnum\z@~
        \tikzi[open~any]
        \bool_gset_false:c {#1_open_left_bool}
        \bool_gset_false:c {#1_open_right_bool}
        \bool_gset_false:c {#1_open_anywhere_bool}
      }
    \bool_if:cT {#1_open_anywhere_bool}
      {
        \global\@topnum\z@~
        \tikzi[open~anywhere ]
        \bool_gset_false:c {#1_open_left_bool}
        \bool_gset_false:c {#1_open_right_bool}
        \bool_gset_false:c {#1_open_any_bool}
      }
}
\ExplSyntaxOff
```

\chapter ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

The standard LaTeX chapter is renewed next. We avoid all the complications of the kernel with different longish sections for chapters and parts. However some more coding is required here.

```
\ExplSyntaxOn
 \renewcommand\chapter {%
    \phd_set_opening{chapter}
%    \bool_if:NT \chapter_open_left_bool  {
%       \cleartoevenpage
%       \global\@topnum\z@~
%       \noindent\ldots open~left\par \tikzi[open~left]
%       \bool_gset_false:N \chapter_open_right_bool
%       \bool_gset_false:N \chapter_open_any_bool
%       \bool_gset_false:N \chapter_open_anywhere_bool
%       }
%    \bool_if:NT \chapter_open_right_bool
%      {
%        \cleardoublepage
%        \global\@topnum\z@~
%        \noindent\ldots open~right\par \tikzi[open~right]
%        \bool_gset_false:N \chapter_open_left_bool
%        \bool_gset_false:N \chapter_open_any_bool
%        \bool_gset_false:N \chapter_open_anywhere_bool
%      }
%    \bool_if:NT \chapter_open_any_bool
%      {
%        \clearpage
%        \global\@topnum\z@~
%        \noindent\ldots open~any\par \tikzi[open~any]
%        \bool_gset_false:N \chapter_open_left_bool
%        \bool_gset_false:N \chapter_open_right_bool
%        \bool_gset_false:N \chapter_open_anywhere_bool
%      }
%    \bool_if:NT \chapter_open_anywhere_bool
%      {
```

```
%        \global\@topnum\z@~
%        \noindent\ldots open~anywhere\par \tikzi[open~anywhere ]
%        \bool_gset_false:N \chapter_open_left_bool
%        \bool_gset_false:N \chapter_open_right_bool
%        \bool_gset_false:N \chapter_open_any_bool
%      }
%
    \thispagestyle{empty}
   %\newpage
    %\bool_show:c {chapter_open_right_bool}
    \start_section:nnnnnnnnn{chapter}%
      {0}  %level check this conflicts with source2e

      {\l__phd_chapter_indent_tl} %indent#2

      {\l__phd_chapter_beforeskip_tl}%before skip#3

      {\l__phd_chapter_afterskip_tl}% after skip#4


%        \setfont@cx
%        {\l__phd_chapter_fontweight_tl}%
%        {\l__phd_chapter_fontfamily_tl}
%        {\l__phd_chapter_fontsize_tl}
%        {\l__phd_chapter_fontshape_tl}%
          %\expandafter\setfontparam@cx\l__phd_chapter_align_tl;%
          %\color{\l__phd_chapter_color_tl}%5
{}

 }%

```

\section ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

The standard LaTeX subsection is renewed next.

```

\if@ltxcompat
\else
  \renewcommand\section{%
    \@startsection{section}%
      {1}%level check this conflicts with source2e

      {\l__phd_section_indent_tl}%indent#2

      {\l__phd_section_beforeskip_tl}%before skip#3

      {\l__phd_section_afterskip_tl}% after skip#4

      {
        \set_font_aux:n {l__phd_section}
%        {\l__phd_section_fontweight_tl}%
%        {\l__phd_section_fontfamily_tl}
%        {\l__phd_section_fontsize_tl}
```

```
%         {\l__phd_section_fontshape_tl}%
%           \expandafter\setfontparam@cx\l__phd_section_align_tl;%
            \l__phd_section_color_tl %5
        }
 }%
%}
\fi
\ExplSyntaxOff
```

\subsection ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

The standard LaTeX subsection is renewed next.

```
\ExplSyntaxOn
\if@ltxcompat
\else
%
 \renewcommand\subsection
    {
      \@startsection{subsection}
        {1}
        {\l__phd_subsection_indent_tl}%indent
        {\l__phd_subsection_beforeskip_tl}%before skip#3
        {\l__phd_subsection_afterskip_tl}% after skip#4

        {
        \set_font_aux:n {l__phd_subsection}
         % \expandafter\setfontparam@cx\l__phd_subsection_align_tl;%
         \l__phd_subsection_color_tl  %5
        }
   }
\fi
\ExplSyntaxOff
%\renewcommand\subsection{\@startsection{subsection}{2}{\z@}%
%                                        {-3.25explus -1ex minus -
.2ex}%
%                                        {1.5ex plus .2ex}%
%                                        {\normalfont\large\bfseries\raggedright}}
%\fi
%
```

\subsubsection ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

The standard LaTeX subsection is renewed next.

```
\ExplSyntaxOn
\if@ltxcompat
\else
\renewcommand{\subsubsection}
{
  \@startsection{subsubsection}%
    {3}%level
    {\l__phd_subsubsection_indent_tl}%indent
    {\l__phd_subsubsection_beforeskip_tl}
    {\l__phd_subsubsection_afterskip_tl}
```

```
    {
      \set_font_aux:n {l__phd_subsubsection}
%%    {\l__phd_subsubsection_fontweight_tl }
%%    {\l__phd_subsubsection_fontfamily_tl}
%%    {\l__phd_subsubsection_fontsize_tl}
%%    {\l__phd_subsubsection_fontshape_tl}
%%      \expandafter\setfontparam@cx
%      %\l__phd_subsubsection_align_tl;
      \l__phd_subsubsection_color_tl
    }
}
\fi
\ExplSyntaxOff
```

I.1.1 Paragraphs

We now deal with paragraphs and subparagraphs, normally termed 'runin' heads, as they produce headings that are inlined with the text that follows. We add hooks, so that later the key mechanism can be used to pick-up values. Although they are termed runins, there is no issue to display them as block.

\paragraph ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

There is a feature in the standard LaTeX classes that a subparagraph is indented by the value of **\parindent**. This also features in memoir but is absent in the KOMA classes. In our defaults we follow the European norm.

Next we provide our definition for paragraphs.

```
\ExplSyntaxOn
%\if@ltxcompat
%\renewcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
%                                    {3.25ex plus1ex minus.2ex}%
%                                    {-1em}%
%                                    {\normalfont\normalsize\bfseries}}
%\else
%\if@ltxcompat
%\else
  \renewcommand\paragraph{
     \@startsection{paragraph}
     {4}%level
     {\l__phd_paragraph_indent_tl}%indent
     {\l__phd_paragraph_beforeskip_tl}%
     {\l__phd_paragraph_afterskip_tl}%
     {
      \set_font_aux:n {l__phd_paragraph}
%      {\l__phd_paragraph_fontweight_tl}%
%     {\l__phd_paragraph_fontfamily_tl}
%     {\l__phd_paragraph_fontsize_tl}
%     {\l__phd_paragraph_fontshape_tl}%
%     \expandafter\setfontparam@cx\l__phd_paragraph_align_tl;%
      \l__phd_paragraph_color_tl
```

```
      }%
    }
%\fi
\ExplSyntaxOff
```

I.1.2 Subparagraph

\subparagraph ⟨*⟩ [⟨*arg1*⟩] {⟨*arg2*⟩}

There is a feature in the standard LaTeX classes that a subparagraph is indented by the value of **\parindent**. This also features in memoir but is absent in the KOMA classes. In our defaults we follow the European norm.

```
\ExplSyntaxOn
\if@ltxcompat
   \renewcommand\subparagraph{\@startsection{subparagraph}{5}{0pt}%
                                          {2ex}%
                                          {-1em}%
                                         {\normalfont\normalsize\bfseries}}
\else
\renewcommand\subparagraph
      {
         \@startsection{subparagraph}
         {5}%level
         {\l__phd_subparagraph_indent_tl}%indent
         {\l__phd_subparagraph_beforeskip_tl}
         {\l__phd_subparagraph_afterskip_tl}
         {
           \set_font_aux:n {l__phd_subparagraph}
           \expandafter\setfontparam@cx
            \l__phd_subparagraph_align_tl;
            \l__phd_subparagraph_color_tl
         }
        }

\fi
\ExplSyntaxOff
```

Default Settings

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

Setting default values

```
\cxset{%
/LSECT
*LSECT|DFLT
    part format                       = traditional,
    chapter title margin-top-width    =  0cm,
    chapter title margin-right-width  =  1cm,
    chapter title margin-bottom-width = 10pt,
    chapter title margin-left-width   = 0pt,
    chapter align                     = left,
    chapter title align               = left, %checked
    chapter name                      = chapter,
    chapter format                    = fashion,
    chapter font-size                 = Huge,
    chapter font-weight               = bold,
    chapter font-family               = sffamily,
    chapter font-shape                = upshape,
    chapter color                     = black,
    chapter number prefix             = ,
    chapter number suffix             = ,
    chapter numbering                 = arabic,
    chapter indent                    = 0pt,
    chapter beforeskip                = -3cm,
    chapter afterskip                 = 30pt,
    chapter afterindent               = off,
    chapter number after              = ,
    chapter arc                       = 0mm,
    chapter background-color          = white,
    chapter afterindent               = off,
    chapter grow left                 = 0mm,
    chapter grow right                = 0mm,
    chapter rounded corners           = northeast,
    chapter shadow                    = fuzzy halo,
    chapter border-left-width         = 0pt,
    chapter border-right-width        = 0pt,
    chapter border-top-width          = 0pt,
    chapter border-bottom-width       = 0pt,
    chapter padding-left-width        = 0pt,
    chapter padding-right-width       = 10pt,
    chapter padding-top-width         = 10pt,
    chapter padding-bottom-width      = 10pt,
    chapter number color              = white,
    chapter label color               = black,
%
    chapter number font-size        = huge,
    chapter number font-weight      = bfseries,
    chapter number font-family      = sffamily,
    chapter number font-shape       = upshape,
    chapter number align            = Centering,
%
    chapter title font-size        = Huge,
    chapter title font-weight      = bold,
    chapter title font-family      = sffamily,
    chapter title font-shape       = upshape,
```

```
    chapter title color            = black,
/LSECT|DFLT
*LSECT
}

/LSECT
*LSECT|DFLT
LSECT\cxset {
    section name                   = Section,
    section format                 = traditional,
    section align                  = Centering,
    section title align            = Centering, %checked
%
    section font-size              = Large,
    section font-weight            = bfseries,
    section font-family            = serif,
    section font-shape             = upshape,
%
    section number font-size       = Large,
    section number font-weight     = bfseries,
    section number font-family     = serif,
    section number font-shape      = upshape,
%
    section title font-size        = Large,
    section title font-weight      = bfseries,
    section title font-family      = serif,
    section title font-shape       = upshape,
%
    section color                  = black,
    section number prefix          = \thechapter.,
    section number suffix          =,
    section numbering              = arabic,
    section indent                 = 0pt,
    section beforeskip             = 3ex,
    section afterskip              = 1.5ex plus .1ex,
    section afterindent            = on,
    section number after           = \quad,
%
    section arc                    = 3pt,
    section background-color       = white,
    section afterindent                = on,
    section grow left                   = 0mm,
    section grow right                 = 0mm,
    section rounded corners        = northeast,
%
    section border-left-width      = 0pt,
    section border-right-width     = 0pt,
    section border-top-width       = 2pt,
    section border-bottom-width    = 2pt,
%
    section padding-left-width     = 0pt,
    section padding-right-width    = 10pt,
    section padding-top-width      = 2pt,
    section padding-bottom-width   = 2pt,
```

```
%
    section title margin-top-width = 2pt,
    section title color            = thesectiontitlecolor,
    section shadow                 = no shadow,
LSECT}
/LSECT|DFLT
*LSECT
```

```
\cxset
  {
/LSECT
*LSECT|DFLT
%% sybsection
    subsection name                  = Subsection,
    subsection format                = hang,
%
    subsection font-size             = large,
    subsection font-weight           = bfseries,
    subsection font-family           = rmfamily,
    subsection font-shape            = upshape,
%
    subsection number font-size      = large,
    subsection number font-weight    = bfseries,
    subsection number font-family    = rmfamily,
    subsection number font-shape     = upshape,
%
    subsection title font-size       = Large,
    subsection title font-weight     = bfseries,
    subsection title font-family     = sffamily,
    subsection title font-shape      = upshape,
    subsection title color           = bgsexy,
%
    subsection color                 = bgsexy,
    subsection numbering             = arabic,
    subsection align                 = Centering, %checked
    subsection title align           = Centering, %checked
    subsection beforeskip            = -3.25explus -
1ex minus -.2ex,
    subsection afterskip             = 1.5ex plus .2ex,
    subsection number prefix         = \thesection.,
    subsection indent                = 0pt,
    subsection number after          = 0pt,
    subsection background-color      = white,
%
    subsection border-left-width     = 0pt,
    subsection border-right-width    = 0pt,
    subsection border-top-width      = 5pt,
    subsection border-bottom-width   = 5pt,
%
    subsection padding-left-width    = 0pt,
    subsection padding-right-width   = 0pt,
    subsection padding-top-width     = 20pt,
    subsection padding-bottom-width  = 20pt,
    subsection shadow                = drop shadow,
```

```
/LSECT|DFLT
*LSECT
  }

\cxset
  {
/LSECT
*LSECT|DFLT
    subsubsection name                    = Subsubsection,
    subsubsection format                  = hang,
    subsubsection background-color        = white, %checked
    subsubsection afterindent             = off,
%
    subsubsection font-family             = rmfamily,
    subsubsection font-size               = large,
    subsubsection font-weight             = bfseries,
    subsubsection font-family             = tiresias,
    subsubsection font-shape              = upshape,
%
    subsubsection font-family             = sffamily,
    subsubsection font-size               = large,
    subsubsection font-weight             = bfseries,
    subsubsection font-family             = tiresias,
    subsubsection font-shape              = upshape,
%
    subsubsection color                   = black,
    subsubsection number prefix           = \thesubsection,
    subsubsection number suffix           = ,
    subsubsection numbering               = arabic,
    subsubsection indent                  = 0pt,
    subsubsection beforeskip              = -3.25explus -1ex minus -.2ex,
    subsubsection afterskip               = 1.5ex plus .2ex,
    subsubsection align                   = center,
    subsubsection title align             = center,
    subsubsection number after     =,
%
    subsubsection border-left-width       = 0pt,
    subsubsection border-right-width      = 0pt,
    subsubsection border-top-width        = 2pt,
    subsubsection border-bottom-width     = 0pt,
%
    subsubsection padding-left-width      = 0pt,
    subsubsection padding-right-width     = 0pt,
    subsubsection padding-top-width       = 20pt,
    subsubsection padding-bottom-width    = 20pt,
    subsubsection shadow                  = no shadow,
%
    subsubsection title font-size         = large,
    subsubsection title font-weight       = bfseries,
    subsubsection title font-family       = serif,
    subsubsection title font-shape        = itshape,
    subsubsection title color             = thesubsectiontitlecolor,
/LSECT|DFLT
```

```
*LSECT
  }

% paragraph
\cxset
  {
/LSECT
*LSECT|DFLT
    paragraph name                = paragraph,
    paragraph format              = inline,
    paragraph name                = paragraph,
    paragraph font-size           = large,
    paragraph font-weight         = bfseries,
    paragraph font-family         = rmfamily,
    paragraph font-shape          = upshape,
    paragraph numbering           = alpha,
    paragraph number prefix       = \thesubsubsection,
    paragraph align               = flushleft,
    paragraph beforeskip          = 3.25ex plus1ex minus.2ex,
    paragraph afterskip           = -1em,
    paragraph indent              = 0pt,
    paragraph number after        = \quad,
    paragraph color               = bgsexy,
    paragraph background-color    = white,
    paragraph shadow              = no shadow,
    paragraph afterindent         = off
/LSECT|DFLT
*LSECT
  }

\cxset
  {
/LSECT
*LSECT|DFLT
    subparagraph name             = subparagraph,
    subparagraph format           = inline,
    subparagraph name             = subparagraph,
    subparagraph font-size        = large,
    subparagraph font-weight      = bfseries,
    subparagraph font-family      = rmfamily,
    subparagraph font-shape       = upshape,
    subparagraph color            = bgsexy,
    subparagraph background-color = bgsexy,
    subparagraph numbering        = none,
    subparagraph align            = flushleft,
    subparagraph beforeskip       = 3.25ex plus1ex minus .2ex,
    subparagraph afterskip        = -1em,
    subparagraph indent           = 0pt,
    subparagraph number after     = ,
    %subparagraph shadow           = off,
/LSECT|DFLT
*LSECT
  }
```

J.1 Styles

```
\cxset{chapter title style/.style= {
       chapter title align = left,}
 }
\cxset{section title style/.style= {
       section title align = Centering,}
 }
\cxset{section title style/.style= {
       section title align = Centering,}
 }
 \cxset{subsection title style/.style= {
       subsection title align = Centering,}
 }
 \cxset{subsubsection title style/.style=
   {
        subsubsection align       = #1,
        subsubsection title align = #1,
   }
 }
 %
 \cxset{%
    subsubsection title style= raggedright
}
```

Tests

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

We prepare a number of tests to verify that all settings work as advertized.

\testsections⟨*void*⟩

In honor of Barbara Beeton all testing commands are in lowercase, but we also provide them in mixed case for the rest of the crowd.

```
% predefined unnumbered headings for compatibility with springer classes
\newcommand{\preface}[1][\prefacename]{\chapter*{#1}\markboth{#1}{#1}}
\newcommand{\foreword}[1][\forewordname]{\chapter*{#1}\markboth{#1}{#1}}

% TODO
\newcommand{\extrachap}[1]{\chapter*{#1}\markboth{#1}{#1}}
\ExplSyntaxOn
\cs_set:Npn \testsections
  {
    \section{Sections}
    \lorem\par
    \subsection{Subsections}
    \lorem\par
    \subsubsection{Subsubsections}
    \lorem\par
    \paragraph {Paragraph}
  }
\cs_set_eq:NN \TestSections\testsections
\ExplSyntaxOff
```

[1]

[1] 1.0,2018/01/26,Converted to DTX file.

Abstract

This is the Ukrainian language module for the datetime2 package. If you want to use the settings in this module you must install it in addition to installing **datetime2**. If you use babel or polyglossia, you will need this module to prevent them from redefining `\today`. The datetime2 useregional setting must be set to "text" or "numeric" for the language styles to be set. Alternatively, you can set the style in the document using `\DTMsetstyle`, but this may be changed by `\dateukrainian` depending on the value of the useregional setting.

The phd-i18n Package

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

One of the primary aims of the package was to simplify the user interface. At the author level, if one has the appropriate stylesheet, nothing needs to be done.

Section L.1

Usage

To set the main language of the document use:

```
%    \cxset{locale german}
%
```

This is best done early in the preamble.

To use any secondary language in the text, there is no need with LuaLaTeX to do anything in the preamble. Just use the appropriate **\text** command.

```
\textgerman{your text here}
```

For longer texts you can use the environment type commands:

```
\begin{ngerman}
\end{ngerman}
```

All locale names can be inputted in multiple ways: a) Using thelowercase name of the language as found in polyglossia or Babel. Using a title case of the language name i.e., **Greek** or **greek**. You can also use the ISO two code or BCF47 codes for greek it would be **el**. If there are dialects you can use **UKenglish**, as in Babel. Remember you only input the main language. If you do not and the package is included it will default to **UKenglish**.

<*package>

Appendix

Implementation

This chapter provides the objectives of the package, provides the methodology to attack the problem and discusses the difficulties encountered in trying to provide a full declarative interface. 21.00002

16

-2.5

-2.1

0.87865

Section A.1

Why?

Babel is a great package, well maintained and support for main stream languages is very good. However, as it supports all the main LaTeX engines, late options are not supported well. This series of packages relies on LuaTeX and unicode input and output.

Section A.2

Specification

1. Integrate well with Babel's latest versions (3.27).
2. Provide translation strings for all languages listed in the Babel and polyglossia packages and extend this to all the unicode languages.
3. Handle date time.
4. Provide useful macros.
5. Set quotation marks for the language.
6. Provide lua code where appropriate.
7. Handle specific numerals, for languages that have their own numerals.
8. Handle text directionality without having to load the bidi package.
9. Handle Asian languages.
10. Rationalize methodology and algorithms as far as possible.
11. Enable traditional typesetting rules for the language. I am not too sure if this belongs here, as at least for sectioning commands this is provided by the sectioning keys, however it makes no harm to re-intoduce it here.

(a) Different enumerate environments (e.g. for French).

(b) Indentation of paragraphs after sections (e.g. French, basque, Ukrainian).

(c) French spacing after punctuation.

(d) Spacing before colons.

12. No active quotes will be provided by default, as we expect the author to enter commands using unicode. However these will be provided as options and be compatible with Babel.

13. Version 1.0 should work with LuaLaTeX only. Higher versions will be adapted to work with LaTeX and XeTeX, as far as this is feasible..

14. Provide a Go or Lua pre-processor utility to reduce the user mark-up for quotations and other similar cases such as spacing before and after `\ldots`.

Section A.3

Data

1. The CLDR data currently at version 33.1 will be used as the data source for translations available in the CLDR specification. A Go program will be developed to download the data and transform it to LaTeX macros.

2. The CLDR does not provide translations for sectioning and other common strings currently required. These will be manually entered.

3. Number formatting will use the CLDR data.

4. Date formatting will use both the CLDR data as well as current conventional data formatting as used in polyglossia and babel.

Section A.4

Preliminaries

Standard file identification. We first announce the package and require that it be used with LaTeX2e.

```
\NeedsTeXFormat{LaTeX2e}[2017/04/15]%
\ProvidesPackage{phd-i18n}[2018/1/13 v1.0 i18N utilities (YL)]%
```

Section A.5

Keys

The keys are defined using the PGF keys package. I diverted from the normal pattern used in other **phd** packages, using a higher number of paths than normal. I did this, as it can give more flexibility to user extensions and also to have almost a one to one relationship with the nesting found in CLDR language files.

We first define a generic command to generate keys for a language tag.

```
\ExplSyntaxOn
\def\setcaptions#1{
\cxset
 {
  locale/#1/captions/refname/.code                              =
    \cs_set:cpn {refname}{##1}
    \cs_set:cpn {greekrefname} {##1},
  locale/#1/captions/abstractname/.code                         =
    \def\abstractname{##1},
  locale/#1/captions/bibname/.code                              =
    \def\bibname{##1},
  locale/#1/captions/prefacename/.code                          =
    \def\prefacename{##1},
  locale/#1/captions/chaptername/.code                          =
    \def\chaptername{\panunicode ##1},
  locale/#1/captions/appendixname/.code    = \cs_set:cpn {appendixname} {##1},
  locale/#1/captions/contentsname/.code    = \cs_set:cpn {contentsname} {##1},
  locale/#1/captions/listfigurename/.code = \cs_set:cpn {listfigurename} {##1},
  locale/#1/captions/listtablename/.code  = \cs_set:cpn {listtablename} {##1},
  locale/#1/captions/indexname/.code      = \cs_set:cpn {indexname} {##1},
  locale/#1/captions/figurename/.code     = \cs_set:cpn {figurename} {##1},
  locale/#1/captions/tablename/.code      = \cs_set:cpn {tablename} {##1},
  locale/#1/captions/partname/.code       = \cs_set:cpn {partname} {##1},
  locale/#1/captions/pagename/.code       = \cs_set:cpn {pagename} {##1},
  locale/#1/captions/seename/.code        = \cs_set:cpn {seename} {##1},
  locale/#1/captions/alsoname/.code       = \cs_set:cpn {alsoname} {##1},
  locale/#1/captions/enclname/.code       = \cs_set:cpn {enclname} {##1},
  locale/#1/captions/ccname/.code         = \cs_set:cpn {ccname} {##1},
  locale/#1/captions/headtoname/.code     = \cs_set:cpn {headtoname} {##1},
  locale/#1/captions/proofname/.code      = \cs_set:cpn {proofname} {##1},
  locale/#1/captions/glossaryname/.code   = \cs_set:cpn {glossaryname} {##1},
```

For dates we have a slightly different approach than Babel and Polyglossia, we just redefine **\today**. So far I don't see the need to define **\date**⟨*language*⟩. If the language is set as main language it will work ok and for others it will work in a group. It will save +- 100 tokens. We can always add it, afterwards just for documentation if necessary.

```
  locale/#1/date/.code  = \cs_set:cpn {today} {##1}
  \cs_set:cpn {date#1}{##1},
```

Subsection A.5.1

Numbers

For numbers we define one to one keys to match the numbers.json of the CLDR specification for the language.

```
  locale/#1/numbers/defaultnumberingsystem/.code         =
    \def\defaultnumberingsystem{##1},
  locale/#1/numbers/othernumberingsystems/.code          =
    \def\othernumberingsystems{##1},
  locale/#1/numbers/minimumgroupdigits/.code             =
    \def\minimumgroupdigits{##1},
  locale/#1/numbers/symbols/decimal/.code                =
    \def\symbolsdecimal{##1},
  locale/#1/numbers/symbols/group/.code                  =
    \def\symbolsgroup{##1},
  locale/#1/numbers/symbols/list/.code                   =
    \def\symbolslist{##1},
  locale/#1/numbers/symbols/percentsign/.code            =
    \def\symbolspermille{##1},
  locale/#1/numbers/symbols/plussign/.code               =
    \def\symbolsplussign{##1},
  locale/#1/numbers/symbols/minussign/.code              =
    \def\symbolsminussign{##1},
  locale/#1/numbers/symbols/exponential/.code            =
    \def\symbolsexponential{##1},
  locale/#1/numbers/symbols/superscriptingexponent/.code =
    \def\symbolssuperscriptingexponent{##1},
  locale/#1/numbers/symbols/permille/.code               =
    \def\symbolsnan{##1},
  locale/#1/numbers/symbols/infinity/.code               =
    \def\symbolsinfinity{##1},
  locale/#1/numbers/symbols/nan/.code                    =
    \def\symbolsnan{##1},
  locale/#1/numbers/symbols/timeseparator/.code          =
    \def\symbolstimeseparator{##1},
%
```

```
 }
 }

\setcaptions{asturian}
\setcaptions{amharic}
\setcaptions{greek}
\setcaptions{german}
\setcaptions{french}
\setcaptions{italian}
\setcaptions{albanian}
\setcaptions{malayalam}
\setcaptions{basque}
\setcaptions{brazil}
\setcaptions{breton}
\setcaptions{bulgarian}
\setcaptions{catalan}
\setcaptions{croatian}
\setcaptions{czech}
\setcaptions{danish}
\setcaptions{dutch} %TODO
\setcaptions{estonian}
\setcaptions{finnish}
\setcaptions{friulan}
\setcaptions{galician}
\setcaptions{icelandic}
\setcaptions{irish}
\setcaptions{latin}
\setcaptions{latvian}
\setcaptions{lithuanian}
\setcaptions{lsorbian}
\setcaptions{magyar}
\setcaptions{marathi}
\setcaptions{nko}
\setcaptions{norsk}
\setcaptions{occitan}
\setcaptions{piedmontese}
\setcaptions{polish}
\setcaptions{portuges}
\setcaptions{romanian}
\setcaptions{romansh}
\setcaptions{samin}
\setcaptions{serbian}
\setcaptions{serbian~cyrillic}
\setcaptions{slovak}
\setcaptions{slovenian}
\setcaptions{swedish}
\setcaptions{tamil}
\setcaptions{telugu}
\setcaptions{turkish}
\setcaptions{turkmen}
\setcaptions{ukrainian}
\setcaptions{usorbian}
\setcaptions{hangul}
\setcaptions{welsh}
```

```
\setcaptions{russian}
```

Section A.6

Retrieving the translation strings

\gettext[⟨*language*⟩]caption name

```
\NewDocumentCommand{\gettext}{o m m}{%
\cxset{locale~#2}
\csname #3name\endcsname}
```

Retrieve the text for chapter in Danish danishchapter danishappendix
Get appendix in Greek greekappendix

Section A.7

Functions generated via scripts

A number of LaTeX 3 functions are generated automatically via a Go program. The Go script obtains data for a particular language tag via the CLDR database and transforms the data into suitable TeX commands. As TeX commands are stored in memory this is faster for execution.

\l_phd_months_wide_language⟨*month*⟩
: Used to print the wide format month for a ⟨*language*⟩.

\l_phd_months_abbreviated_language⟨*month*⟩
: Used to print the wide format month for a ⟨*language*⟩.

\l_phd_months_narrow_language⟨*month*⟩
: Used to print the wide format month for a ⟨*language*⟩.

Section A.8

Asturian

Astur-Leonese is the historical language of Asturias, portions of the Spanish provinces of León and Zamora and the area surrounding Miranda do Douro in northeastern Portugal.[8] Like the other Romance languages of the Iberian peninsula, it evolved from Vulgar Latin during the early Middle Ages. Asturian was closely linked with the Kingdom of Asturias (718–910) and the ensuing Leonese kingdom. The language had contributions from pre-Roman languages spoken by the Astures, an Iberian Celtic tribe, and the post-Roman Germanic languages of the Visigoths and Suevi. CLDR language code = ast

```
\cxset{locale~asturian/.style = {
  locale/asturian/captions/refname        = Referencies,
  locale/asturian/captions/abstractname   = Sumariu,
  locale/asturian/captions/bibname        = Bibliografía,
  locale/asturian/captions/prefacename    = Entamu,
  locale/asturian/captions/chaptername    = Capítulu,
  locale/asturian/captions/appendixname   = Apéndiz,
  locale/asturian/captions/contentsname   = Conteníu,
  locale/asturian/captions/listfigurename = Llista~de~figures,
  locale/asturian/captions/listtablename  = Llista~de~tables,
  locale/asturian/captions/indexname      = Índiz,
  locale/asturian/captions/figurename     = Figura,
  locale/asturian/captions/tablename      = Tabla,
  locale/asturian/captions/partname       = Parte,
  locale/asturian/captions/pagename       = Páxina,
  locale/asturian/captions/seename        = ver,
  locale/asturian/captions/alsoname       = ver~tamién,
  locale/asturian/captions/enclname       = incl.,
  locale/asturian/captions/ccname         = cc,
  locale/asturian/captions/headtoname     = Pa,
  locale/asturian/captions/proofname      = Demostración,
  locale/asturian/captions/glossaryname   = Glosariu,
  locale/asturian/date=\number\day~\ifcase\month\or
    de~xineru\or de~febreru\or de~marzu\or d'abril\or de~mayu\or de~xunu\or
    de~xunetu\or d'agostu\or de~setiembre\or d'ochobre\or de~payares\or
    d'avientu\fi\space de~\number\year,
}}
\cxset{locale~Asturian/.alias = locale~asturian}
```

Asturian

Section A.9

Amharic

CLDR is am

```
\cxset{locale~amharic/.style = {
  locale/amharic/captions/refname        = ~~,
  locale/amharic/captions/abstractname   = ~,
  locale/amharic/captions/bibname        = ~,
  locale/amharic/captions/prefacename    = ,
  locale/amharic/captions/chaptername    = ,
  locale/amharic/captions/appendixname   = ,
  locale/amharic/captions/contentsname   = ,
  locale/amharic/captions/listfigurename = ~,
  locale/amharic/captions/listtablename  = ~,
  locale/amharic/captions/indexname      = ~,
  locale/amharic/captions/figurename     = ,
  locale/amharic/captions/tablename      = ,
  locale/amharic/captions/partname       =  ,
  locale/amharic/captions/pagename       = ,
  locale/amharic/captions/seename        = ,
  locale/amharic/captions/alsoname       = ~,
  locale/amharic/captions/enclname       = ,
  locale/amharic/captions/ccname         = ,
  locale/amharic/captions/headtoname     = ,
  locale/amharic/captions/proofname      = ,
  locale/amharic/captions/glossaryname   = ~,
  locale/amharic/date=\number\day\space\ifcase\month\or
    de~xineru\or de~febreru\or de~marzu\or d'abril\or de~mayu\or de~xunu\or
    de~xunetu\or d'agostu\or de~setiembre\or d'ochobre\or de~payares\or
    d'avientu\fi\space de~\number\year,
}}

\cxset{locale~Amharic/.alias = locale~amharic}
```

Amharic

Section A.10

Greek

```
\cs_set:Npn \l_phd_months_wide_greek #1 {
  \if_case:w #1
    \or:
    \or:
    \or:
    \or: π
    \or:
    \or:
    \or:
    \or:
    \or: π
    \or:
    \or:
    \or:
  \fi:
}
\cxset{locale~greek/.style = {
  locale/greek/captions/refname        = ,
  locale/greek/captions/abstractname   = ,
  locale/greek/captions/bibname        = ,
  locale/greek/captions/prefacename    = ,
  locale/greek/captions/chaptername    = ,
  locale/greek/captions/appendixname   = ,
  locale/greek/captions/contentsname   = ,
  locale/greek/captions/listfigurename = ~,
  locale/greek/captions/listtablename  = ~π,
  locale/greek/captions/indexname      = ,
  locale/greek/captions/figurename     = ,
  locale/greek/captions/tablename      = ,
  locale/greek/captions/partname       = ,
  locale/greek/captions/pagename       = ,
  locale/greek/captions/seename        = π,
  locale/greek/captions/alsoname       = π~π,
  locale/greek/captions/enclname       = ,
  locale/greek/captions/ccname         = π,
  locale/greek/captions/headtoname     = ,
  locale/greek/captions/proofname      = π,
  locale/greek/captions/glossaryname   = ,
  locale/greek/date = {\number\day\space%
     \l_phd_months_wide_greek {\month}
      \space\number\year},
}}
\cxset{locale~Greek/.alias = locale~Greek}
```

Greek

Section A.11

German

```
\cs_set:Npn \l_phd_months_wide_german #1 {
  \if_case:w #1
    \or: Januar
    \or: Februar
    \or: März
    \or: April
    \or: Mai
    \or: Juni
    \or: Juli
    \or: August
    \or: September
    \or: Oktober
    \or: November
    \or: Dezember
  \fi:
}
\cxset{locale~german/.style = {
  locale/german/captions/refname        = Literatur,
  locale/german/captions/abstractname   = Zusammenfassung,
  locale/german/captions/bibname        = Literaturverzeichnis,
  locale/german/captions/prefacename    = Vorwort,
  locale/german/captions/chaptername    = Kapitel,
  locale/german/captions/appendixname   = Anhang,
  locale/german/captions/contentsname   = Inhaltsverzeichnis,
  locale/german/captions/listfigurename = Abbildungsverzeichnis,
  locale/german/captions/listtablename  = Tabellenverzeichnis,
  locale/german/captions/indexname      = Index,
  locale/german/captions/figurename     = Abbildung,
  locale/german/captions/tablename      = Tabelle,
  locale/german/captions/partname       = Teil,
  locale/german/captions/pagename       = Seite,
  locale/german/captions/seename        = siehe,
  locale/german/captions/alsoname       = siehe~auch,
  locale/german/captions/enclname       = Anlage(n),
  locale/german/captions/ccname         = Verteiler,
  locale/german/captions/headtoname     = An,
  locale/german/captions/proofname      = Beweis,
  locale/german/captions/glossaryname   = Glossar,
  locale/german/date = {\number\day.%
    \space
    \l_phd_months_wide_german{\month}
    \space \number\year},
}}
\cxset{locale~German/.alias=locale~german}
```

German

Section A.12

French

Subsection A.12.1

French months

Months for French are defined as per CLDR.

```
\cs_set:Npn \l_phd_months_wide_french #1 {
  \if_case:w #1
     \or: janvier
     \or: février
     \or: mars
     \or: avril
     \or: mai
     \or: juin
     \or: juillet
     \or: août
     \or: septembre
     \or: octobre
     \or: novembre
     \or: décembre
  \fi:
}
%
\cs_set:Npn \l_phd_months_wide_abbreviated #1 {
  \if_case:w #1
     \or: janv.,
     \or: févr.,
     \or: mars,
     \or: avr.,
     \or: mai,
     \or: juin,
     \or: juil.,
     \or: août,
     \or: sept.,
     \or: oct.,
     \or: nov.,
     \or: déc.,
  \fi:
}
%
\cxset{locale~french/.style = {
  locale/french/captions/refname        = Références,
  locale/french/captions/abstractname   = Résumé,
  locale/french/captions/bibname        = Bibliographie,
  locale/french/captions/prefacename    = Préface,
  locale/french/captions/chaptername    = Chapitre,
  locale/french/captions/appendixname   = Annexe,
  locale/french/captions/contentsname   = Table~des~matières,
```

```
  locale/french/captions/listfigurename = Table~des~figures,
  locale/french/captions/listtablename  = Liste~des~tableaux,
  locale/french/captions/indexname      = Index ,
  locale/french/captions/figurename     = \textsc{Fig.},
  locale/french/captions/tablename      = \textsc{Tab.} ,
  locale/french/captions/partname       = ,
  locale/french/captions/pagename       = page,
  locale/french/captions/seename        = \emph{voir},
  locale/french/captions/alsoname       = \emph{voir~aussi} ,
  locale/french/captions/enclname       = P.~J.,
  locale/french/captions/ccname         = Copie~à ,
  locale/french/captions/headtoname     = {} ,
  locale/french/captions/proofname      = Démonstration,
  locale/french/captions/glossaryname   = ,
  locale/french/date =
      {
        \ifx\ier\undefined\def\ier{er}\fi
        \ifnum\day=1\relax~1\ier%
        \else \number\day\fi
        \space
        \l_phd_months_wide_french {\month}
        \space
        \number\year
      },
}}

\cxset{locale~French/.alias = locale~french}
```

French

Section A.13

Italian

```
\cs_set:Npn \l_phd_months_wide_italian #1 {
  \if_case:w #1
     \or: gennaio
     \or: febbraio
     \or: marzo
     \or: aprile
     \or: maggio
     \or: giugno
     \or: luglio
     \or: agosto
     \or: settembre
     \or: ottobre
     \or: novembre
     \or: dicembre
```

```
  \fi:
}
\cxset{locale~italian/.style = {
  locale/italian/captions/refname        = Riferimenti bibliografici,
  locale/italian/captions/abstractname   = Sommario ,
  locale/italian/captions/bibname        = Bibliografia ,
  locale/italian/captions/prefacename    = Prefazione ,
  locale/italian/captions/chaptername    = Capitolo,
  locale/italian/captions/appendixname   = Appendice ,
  locale/italian/captions/contentsname   = Indice ,
  locale/italian/captions/listfigurename = Elenco~delle~figure,
  locale/italian/captions/listtablename  = Elenco~delle~tabelle,
  locale/italian/captions/indexname      = Indice~analitico,
  locale/italian/captions/figurename     = Figura,
  locale/italian/captions/tablename      = Tabella,
  locale/italian/captions/partname       = Parte ,
  locale/italian/captions/pagename       = Pag. , %in Italian abbreviation is preferred
  locale/italian/captions/seename        = vedi ,
  locale/italian/captions/alsoname       = vedi~anche,
  locale/italian/captions/enclname       = Allegati,
  locale/italian/captions/ccname         = e~p.~c. ,
  locale/italian/captions/headtoname     = Per,
  locale/italian/captions/proofname      = Dimostrazione,
  locale/italian/captions/glossaryname   = Glossario,
  locale/italian/date = {\number\day\space
    \l_phd_months_wide_italian {\month}
   \space\number\year},
}}
\cxset{locale~Italian/.alias = locale~italian}
```

Italian

Section A.14

Albanian

```
\cs_set:Npn \l_phd_months_wide_albanian #1 {
  \if_case:w #1
    \or: Janar
    \or: Shkurt
    \or: Mars
    \or: Prill
    \or: Maj
    \or: Qershor
    \or: Korrik
    \or: Gusht
    \or: Shtator
    \or: Tetor
    \or: Nëntor
```

```
    \or: Dhjetor
  \fi:
}
\cxset{locale~albanian/.style = {
  locale/albanian/captions/refname        = Referencat,
  locale/albanian/captions/abstractname   = Përmbledhja ,
  locale/albanian/captions/bibname        = Bibliografia ,
  locale/albanian/captions/prefacename    = Parathenia,
  locale/albanian/captions/chaptername    = Kapitulli,
  locale/albanian/captions/appendixname   = Shtesa,
  locale/albanian/captions/contentsname   = Përmbajta,
  locale/albanian/captions/listfigurename = Figurat,
  locale/albanian/captions/listtablename  = Tabelat,
  locale/albanian/captions/indexname      = Indeksi,
  locale/albanian/captions/figurename     = Figura,
  locale/albanian/captions/tablename      = Tabela,
  locale/albanian/captions/partname       = Pjesa,
  locale/albanian/captions/pagename       = Faqe,
  locale/albanian/captions/seename        = shiko,
  locale/albanian/captions/alsoname       = shiko~dhe,
  locale/albanian/captions/enclname       = ,
  locale/albanian/captions/ccname         = ,
  locale/albanian/captions/headtoname     = ,
  locale/albanian/captions/proofname      = Vërtetim,
  locale/albanian/captions/glossaryname   = Përhasja~e~Fjalëve ,
  locale/albanian/date=\number\day\space
  \l_phd_months_wide_albanian {\month}
   \space \number\year,
}}
```

Section A.15

Malayalam

```
\cxset{locale~malayalam/.style = {
  locale/malayalam/captions/refname = ,
  locale/malayalam/captions/abstractname   = ,
  locale/malayalam/captions/bibname        = ,
  locale/malayalam/captions/prefacename    = ,
  locale/malayalam/captions/chaptername    = \panunicode ,
  locale/malayalam/captions/appendixname   = ,
  locale/malayalam/captions/contentsname   = \panunicode ,
  locale/malayalam/captions/listfigurename = ,
  locale/malayalam/captions/listtablename  = ~,
  locale/malayalam/captions/indexname      = ,
  locale/malayalam/captions/figurename     = ,
  locale/malayalam/captions/tablename      = ,
  locale/malayalam/captions/partname       = ,
```

```
  locale/malayalam/captions/pagename       = ,
  locale/malayalam/captions/seename        = ,
  locale/malayalam/captions/alsoname       = ~,
  locale/malayalam/captions/enclname       = ,
  locale/malayalam/captions/ccname         = ,
  locale/malayalam/captions/headtoname     = ,
  locale/malayalam/captions/proofname      = ,
  locale/malayalam/captions/glossaryname   = ,
  locale/malayalam/date                    = {
  \panunicode\number\year\space\ifcase\month\or
     \or
     \or
     \or
     \or
     \or
     \or
     \or
     \or
     \or
     \or
     \or
     \fi
     \space\number\day},
}}
```

malayalam

Section A.16

Russian

```
\cs_set:Npn \l_phd_months_wide_russian #1 {
  \if_case:w #1
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
     \or:
  \fi:
}
\cxset{locale~russian/.style = {
  locale/russian/captions/refname        = ~,
```

```
  locale/russian/captions/abstractname   = ,
  locale/russian/captions/bibname        = ,
  locale/russian/captions/prefacename    = ,
  locale/russian/captions/chaptername    = ,
  locale/russian/captions/appendixname

   = ,
  locale/russian/captions/contentsname   = ,
  locale/russian/captions/listfigurename = ~,
  locale/russian/captions/listtablename  = ~,
  locale/russian/captions/indexname      = ~,
  locale/russian/captions/figurename     = .,
  locale/russian/captions/tablename      = ,
  locale/russian/captions/partname       = ,
  locale/russian/captions/pagename       = .,
  locale/russian/captions/seename        = .,
  locale/russian/captions/alsoname       = .~,
  locale/russian/captions/enclname       = .,
  locale/russian/captions/ccname         = .,
  locale/russian/captions/headtoname     = .,
  locale/russian/captions/proofname      = ,
  locale/russian/captions/glossaryname = ,
  locale/russian/date =
    {\number\day%
      \space
       \l_phd_months_wide_russian {\month}
      \space \number\year\space .},
  }
}
```

russian

Section A.17

Basque

The Basque language definitions are in line with those of Babel and Polyglossia. Sadly we do not have any miller's numbers yet. See Languages Monograph for more details.

```
\cs_set:Npn \l_phd_months_wide_basque #1 {
  \if_case:w #1
    \or: urtarrilaren
    \or: otsailaren
    \or: martxoaren
    \or: apirilaren
    \or: maiatzaren
    \or: ekainaren
    \or: uztailaren
```

```
    \or: abuztuaren
    \or: irailaren
    \or: urriaren
    \or: azaroaren
    \or: abenduaren
  \fi:
}
\cxset{locale~basque/.style = {
  locale/basque/captions/refname        = Erreferentziak,
  locale/basque/captions/abstractname   = Laburpena,
  locale/basque/captions/bibname        = Bibliografia,
  locale/basque/captions/prefacename    = Hitzaurrea,
  locale/basque/captions/chaptername    = Kapitulua,
  locale/basque/captions/appendixname   = Eranskina,
  locale/basque/captions/contentsname   = Gaien~Aurkibidea,
  locale/basque/captions/listfigurename = Irudien~Zerrenda,
  locale/basque/captions/listtablename  = Taulen Zerrenda,
  locale/basque/captions/indexname      = Kontzeptuen Aurkibidea,
  locale/basque/captions/figurename     = Irudia,
  locale/basque/captions/tablename      = Taula,
  locale/basque/captions/partname       = Atala,
  locale/basque/captions/pagename       = Orria ,
  locale/basque/captions/seename        = Ikusi,
  locale/basque/captions/alsoname       = {Ikusi,~halaber}, % has a comma!
  locale/basque/captions/enclname       = Erantsia,
  locale/basque/captions/ccname         = Kopia,
  locale/basque/captions/headtoname     = Nori,
  locale/basque/captions/proofname      = Frogapena,
  locale/basque/captions/glossaryname   = Glosarioa,
  locale/basque/date = {\number\year.eko\space
  \l_phd_months_wide_basque{\month}
  \space\number\day},
}}
```

basque

Section A.18

Brazil

```
\cs_set:Npn \l_phd_months_wide_brazil #1
  {
    \if_case:w #1
      \or: janeiro
      \or: fevereiro
      \or: março
      \or: abril
```

```
      \or: maio
      \or: junho
      \or: julho
      \or: agosto
      \or: setembro
      \or: outubro
      \or: novembro
      \or: dezembro
    \fi:
  }
\cxset{locale~brazil/.style = {
  locale/brazil/captions/refname        = Referências,
  locale/brazil/captions/abstractname   = Resumo,
  locale/brazil/captions/bibname        = Referências~Bibliográficas,
  locale/brazil/captions/prefacename    = Prefácio,
  locale/brazil/captions/chaptername    = Capitulo,
  locale/brazil/captions/appendixname   = Apêndice,
  locale/brazil/captions/contentsname   = Sumário,
  locale/brazil/captions/listfigurename = Lista~de~Figuras,
  locale/brazil/captions/listtablename  = Lista~de~Tabelas,
  locale/brazil/captions/indexname      = Índice~Remissivo,
  locale/brazil/captions/figurename     = Figura,
  locale/brazil/captions/tablename      = Tabela,
  locale/brazil/captions/partname       = Parte,
  locale/brazil/captions/pagename       = Página,
  locale/brazil/captions/seename        = veja,
  locale/brazil/captions/alsoname       = veja~também,
  locale/brazil/captions/enclname       = Anexo,
  locale/brazil/captions/ccname         = Cópia~para,
  locale/brazil/captions/headtoname     = Para,
  locale/brazil/captions/proofname      = Demonstração,
  locale/brazil/captions/glossaryname   = Glossário,
  locale/brazil/date = {\number\day\space de\space\ifcase\month\or
      janeiro\or fevereiro\or março\or abril\or maio\or junho\or
      julho\or agosto\or setembro\or outubro\or novembro\or dezembro%
      \fi\space de\space\number\year},
}}
```

brazil

Section A.19

Breton

```
\cs_set:Npn \l_phd_months_wide_breton #1
  {
    \if_case:w #1
     \or: Genver
     \or: C'hwevrer
```

```
      \or: Meurzh
      \or: Ebrel
      \or: Mae
      \or: Mezheven
      \or: Gouere
      \or: Eost
      \or: Gwengolo
      \or: Here
      \or: Du
      \or: Kerzu
    \fi:
  }
\cxset{locale~breton/.style = {
  locale/breton/captions/refname        = Daveennoù,
  locale/breton/captions/abstractname   = Dvierrañ,
  locale/breton/captions/bibname        = Lennadurezh,
  locale/breton/captions/prefacename    = Rakskrid,
  locale/breton/captions/chaptername    = Pennad,
  locale/breton/captions/appendixname   = Stagadenn,
  locale/breton/captions/contentsname   = Taolenn,
  locale/breton/captions/listfigurename = Listenn~ar~Figurennoù,
  locale/breton/captions/listtablename  = Listenn~an~taolennoù,
  locale/breton/captions/indexname      = Meneger,
  locale/breton/captions/figurename     = Figurenn,
  locale/breton/captions/tablename      = Taolenn,
  locale/breton/captions/partname       = Lodenn,
  locale/breton/captions/pagename       = Pajenn,
  locale/breton/captions/seename        = Gwelout,
  locale/breton/captions/alsoname       = Gwelout~ivez,
  locale/breton/captions/enclname       = Dielloù~kevret,
  locale/breton/captions/ccname         = Eilskrid~da,
  locale/breton/captions/headtoname     = evit,
  locale/breton/captions/proofname      = Proof,
  locale/breton/captions/glossaryname   = Glossary,
  locale/breton/date =
    {\ifnum\day=1\relax 1\/\textsuperscript{añ}\else
    \number\day\fi \space a\space viz\space
    \l_phd_months_wide_breton {\month}
    \space\number\year},
  }
}
```

Section A.20

Bulgarian

```
\cs_set:Npn \l_phd_months_wide_bulgarian #1
  {
```

```
    \if_case:w #1
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
      \or:
    \fi:
  }
\cxset{locale~bulgarian/.style = {
  locale/bulgarian/captions/refname        = ,
  locale/bulgarian/captions/abstractname   = ,
  locale/bulgarian/captions/bibname        = ,
  locale/bulgarian/captions/prefacename    = ,
  locale/bulgarian/captions/chaptername    = ,
  locale/bulgarian/captions/appendixname   = ,
  locale/bulgarian/captions/contentsname   = ,
  locale/bulgarian/captions/listfigurename = ~~,
  locale/bulgarian/captions/listtablename  = ~~,
  locale/bulgarian/captions/indexname      = ~,
  locale/bulgarian/captions/figurename     = ,
  locale/bulgarian/captions/tablename      = ,
  locale/bulgarian/captions/partname       = ,
  locale/bulgarian/captions/pagename       = .,
  locale/bulgarian/captions/seename        = .,
  locale/bulgarian/captions/alsoname       = .\~~,
  locale/bulgarian/captions/enclname       = ,
  locale/bulgarian/captions/ccname         = ,
  locale/bulgarian/captions/headtoname     = {},
  locale/bulgarian/captions/proofname      = Proof,
  locale/bulgarian/captions/glossaryname   = Glossary,
  locale/bulgarian/date = {\number\day\space
       \l_phd_months_wide_bulgarian {\month}
       \space\number\year\space .},
}}
```

bulgarian

Section A.21

Catalan

```
\cs_set:Npn \l_phd_months_wide_catalan #1
```

```
  {
    \if_case:w #1
     \or: de~gener
     \or: de~febrer
     \or: de~març
     \or: d'abril
     \or: de~maig
     \or: de~juny
     \or: de~juliol
     \or: d'agost
     \or: de setembre
     \or: d'octubre
     \or: de~novembre
     \or: de~desembre
    \fi:
  }
\cxset{locale~catalan/.style = {
  locale/catalan/captions/refname        = Referències,
  locale/catalan/captions/abstractname   = Resum,
  locale/catalan/captions/bibname        = Bibliografia,
  locale/catalan/captions/prefacename    = Pròleg,
  locale/catalan/captions/chaptername    = Capítol,
  locale/catalan/captions/appendixname   = Apèndix,
  locale/catalan/captions/contentsname   = Índex,
  locale/catalan/captions/listfigurename = Índex~de~figures,
  locale/catalan/captions/listtablename  = Índex~de~taules,
  locale/catalan/captions/indexname      = Índex~alfabètic,
  locale/catalan/captions/figurename     = Figura,
  locale/catalan/captions/tablename      = Taula,
  locale/catalan/captions/partname       = Part,
  locale/catalan/captions/pagename       = Pàgina ,
  locale/catalan/captions/seename        = Vegeu,
  locale/catalan/captions/alsoname       = Vegeu~també ,
  locale/catalan/captions/enclname       = Adjunt,
  locale/catalan/captions/ccname         = Còpies~a,
  locale/catalan/captions/headtoname     = A,
  locale/catalan/captions/proofname      = Demostració,
  locale/catalan/captions/glossaryname   = Glossari,
  locale/catalan/date ={\number\day\space
    \l_phd_months_wide_catalan {\month}
    \space de~\number\year} ,
}}
\cxset{locale~Catalan/.alias = locale~catalan}
```

Catalan

Section A.22

Croatian

```
\cs_set:Npn \l_phd_months_wide_croatian #1
  {
    \if_case:w #1
     \or: siječnja
     \or: veljače
     \or: ožujka
     \or travnja
     \or svibnja
     \or lipnja
     \or srpnja
     \or kolovoza
     \or rujna
     \or listopada
     \or studenoga
     \or prosinca
    \fi:
  }
\cxset{locale~croatian/.style = {
  locale/croatian/captions/refname        = Literatura,
  locale/croatian/captions/abstractname   = Sažetak,
  locale/croatian/captions/bibname        = Bibliografija,
  locale/croatian/captions/prefacename    = Predgovor,
  locale/croatian/captions/chaptername    = Poglavlje,
  locale/croatian/captions/appendixname   = Dodatak,
  locale/croatian/captions/contentsname   = Sadržaj,
  locale/croatian/captions/listfigurename = Popis~slika,
  locale/croatian/captions/listtablename  = Popis~tablica,
  locale/croatian/captions/indexname      = Kazalo,
  locale/croatian/captions/figurename     = Slika,
  locale/croatian/captions/tablename      = Tablica,
  locale/croatian/captions/partname       = Dio,
  locale/croatian/captions/pagename       = Stranica,
  locale/croatian/captions/seename        = Vidjeti,
  locale/croatian/captions/alsoname       = Također~vidjeti,
  locale/croatian/captions/enclname       = Prilozi,
  locale/croatian/captions/ccname         = Kopija,
  locale/croatian/captions/headtoname     = Prima,
  locale/croatian/captions/proofname      = Dokaz,
  locale/croatian/captions/glossaryname   = Pojmovnik,
  locale/croatian/date ={\number\day.\space
  \l_phd_months_wide_croatian {\month}
  \space \number\year.},
}}
```

Section A.23

Czech

```
\cs_set:Npn \l_phd_months_wide_czech #1{
  \if_case:w #1
    \or: ledna
    \or: února
    \or: března
    \or: dubna
    \or: května
    \or: června
    \or: července
    \or: srpna
    \or: září
    \or: října
    \or: listopadu
    \or: prosince
  \fi:
}
\cxset{locale~czech/.style = {
  locale/czech/captions/refname        = Referències,
  locale/czech/captions/abstractname   = Resum,
  locale/czech/captions/bibname        = Bibliografia,
  locale/czech/captions/prefacename    = Pròleg,
  locale/czech/captions/chaptername    = Kapitola,
  locale/czech/captions/appendixname   = Apèndix,
  locale/czech/captions/contentsname   = Obsah,
  locale/czech/captions/listfigurename = Índex~de~figures,
  locale/czech/captions/listtablename  = Índex~de~taules,
  locale/czech/captions/indexname      = Índex~alfabètic,
  locale/czech/captions/figurename     = Figura,
  locale/czech/captions/tablename      = Taula,
  locale/czech/captions/partname       = Part,
  locale/czech/captions/pagename       = Pàgina,
  locale/czech/captions/seename        = Vegeu,
  locale/czech/captions/alsoname       = Vegeu~també,
  locale/czech/captions/enclname       = Adjunt,
  locale/czech/captions/ccname         = Còpies~a,
  locale/czech/captions/headtoname     = A,
  locale/czech/captions/proofname      = Demostració,
  locale/czech/captions/glossaryname   = Glossari,
  locale/czech/date =
    {\number\day.\space
     \l_phd_months_wide_czech {\month}
    \space \number\year},
    }
}
```

Section A.24

Danish

```
\cs_set:Npn \l_phd_months_wide_danish #1{
  \if_case:w #1
    \or: januar
    \or: februar
    \or: marts
    \or: april
    \or: maj
    \or: juni
    \or: juli
    \or: august
    \or: september
    \or: oktober
    \or: november
    \or: december
  \fi:
}
\cxset{locale~danish/.style = {
  locale/danish/captions/refname        = Litteratur,
  locale/danish/captions/abstractname   = Resumé,
  locale/danish/captions/bibname        = Litteratur,
  locale/danish/captions/prefacename    = Forord,
  locale/danish/captions/chaptername    = Kapitel,
  locale/danish/captions/appendixname   = Bilag,
  locale/danish/captions/contentsname   = Indhold,
  locale/danish/captions/listfigurename = Figurer,
  locale/danish/captions/listtablename  = Tabeller,
  locale/danish/captions/indexname      = Indeks,
  locale/danish/captions/figurename     = Figur,
  locale/danish/captions/tablename      = Tabel,
  locale/danish/captions/partname       = Del,
  locale/danish/captions/pagename       = Side,
  locale/danish/captions/seename        = Se,
  locale/danish/captions/alsoname       = Se~også,
  locale/danish/captions/enclname       = Vedlagt,
  locale/danish/captions/ccname         = Kopi~til,
  locale/danish/captions/headtoname     = Til,
  locale/danish/captions/proofname      = Bevis,
  locale/danish/captions/glossaryname   = Gloseliste ,
  locale/danish/date ={\number\day.\space
    \l_phd_months_wide_danish {\month}
    \space\number\year},
}}
```

Section A.25

Estonian

See estonian local page for standards.

```
\cs_set:Npn \l_phd_months_wide_estonian #1{
  \if_case:w #1
    \or: jaanuar
    \or: veebruar
    \or: märts
    \or: aprill
    \or: mai
    \or: juuni
    \or: juuli
    \or: august
    \or: september
    \or: oktoober
    \or: november
    \or: detsember
  \fi:
}
\cxset{locale~estonian/.style = {
  locale/estonian/captions/refname = Viited,
  locale/estonian/captions/abstractname   = Kokkuvõte,
  locale/estonian/captions/bibname        = Kirjandus,
  locale/estonian/captions/prefacename    = Sissejuhatus,
  locale/estonian/captions/chaptername    = Peatükk,
  locale/estonian/captions/appendixname   = Lisa,
  locale/estonian/captions/contentsname   = Sisukord,
  locale/estonian/captions/listfigurename = Joonised,
  locale/estonian/captions/listtablename  = Tabelid,
  locale/estonian/captions/indexname      = Indeks,
  locale/estonian/captions/figurename     = Joonis ,
  locale/estonian/captions/tablename      = Tabel,
  locale/estonian/captions/partname       = Osa,
  locale/estonian/captions/pagename       = Lk.,
  locale/estonian/captions/seename        = vt.,
  locale/estonian/captions/alsoname       = vt.~ka,
  locale/estonian/captions/enclname       = Lisa(d),
  locale/estonian/captions/ccname         = Koopia(d),
  locale/estonian/captions/headtoname     = ,
  locale/estonian/captions/proofname      = Korrektuur,
  locale/estonian/captions/glossaryname   = , %unknown
  locale/estonian/date =
   {\number\day.\space
      \l_phd_months_wide_estonian{\month}
      \space\number\year.\space a.
   },
}}
\cxset{locale~esti/.alias=locale~estonian,
       esti/.alias=locale~estonian,
       locale~Greek/.alias=locale~greek}
```

Section A.26

Finnish

The data for Finnish has been collected from Polyglossia, Babel, Translator, CLDR files and IBM. It is missing at this stage .

```
\cs_set:Npn \l_phd_months_abbreviated_finnish#1{
  \if_case:w #1
    \or: tammi
    \or: helmi
    \or: maalis
    \or: huhti
    \or: touko
    \or: kesä
    \or: heinä
    \or: eloa
    \or: syys
    \or: loka
    \or: marras
    \or: joulu
  \fi:
}
```

I am not too sure if the below are correct. They have been copied from the relevant CLDR file, but as we can see these abbreviations will result in ambiguous date strings. The example at IBM [1] uses two character abbreviations, but does not list all of them.

```
\cs_set:Npn \l_phd_months_narrow_finnish#1{
  \if_case:w #1
    \or: T
    \or: H
    \or: M
    \or: H
    \or: T
    \or: K
    \or: H
    \or: E
    \or: S
    \or: L
    \or: M
    \or: J
  \fi:
}
```

The wide months are used as the default.

[1] See IBM Knowledge Center

```
\cs_set:Npn \l_phd_months_wide_finnish#1{
  \if_case:w #1
    \or: tammikuuta
    \or: helmikuuta
    \or: maaliskuuta
    \or: huhtikuuta
    \or: toukokuuta
    \or: kesäkuuta
    \or: heinäkuuta
    \or: elokuuta
    \or: syyskuuta
    \or: lokakuuta
    \or: marraskuuta
    \or: joulukuuta
  \fi:
}
```

Subsection A.26.1

Lists

The Finnish alphabet has three extra letters Å, Ä and Ö. The “Swedish o” is redundant in Finnish, but is used for writing Finland-Swedish proper names.

```
\cs_new:Npn \int_to_Alph_finnish:n #1
 {
 \int_to_symbols:nnn {#1} { 29 }
 {
  { 1 } { A }
  { 2 } { B }
  { 3 } { C }
  { 4 } { D }
  { 5 } { E }
  { 6 } { F }
  { 7 } { G }
  { 8 } { H }
  { 9 } { I }
  { 10 } { J }
  { 11 } { K }
  { 12 } { L }
  { 13 } { M }
  { 14 } { N }
  { 15 } { O }
  { 16 } { P }
  { 17}  { Q }
  { 18 } { R }
  { 19 } { S }
```

```
  { 20 } { T }
  { 21 } { U }
  { 22 } { V }
  { 23 } { W }
  { 24 } { X }
  { 25 } { Y }
  { 26 } { Z }
  { 27 } { Å }
  { 28 } { Ä }
  { 29 } { Ö }
 }
}
\cs_new:Npn \int_to_alph_finnish:n #1
 {
 \int_to_symbols:nnn {#1} { 29 }
 {
  { 1 } { a }
  { 2 } { b }
  { 3 } { c }
  { 4 } { d }
  { 5 } { e }
  { 6 } { f }
  { 7 } { g }
  { 8 } { h }
  { 9 } { i }
  { 10 } { j }
  { 11 } { k }
  { 12 } { l }
  { 13 } { m }
  { 14 } { n }
  { 15 } { o }
  { 16 } { p }
  { 17}  { q }
  { 18 } { r }
  { 19 } { s }
  { 20 } { t }
  { 21 } { u }
  { 22 } { v }
  { 23 } { w }
  { 24 } { x }
  { 25 } { y }
  { 26 } { z }
  { 27 } { å }
  { 28 } { ä }
  { 29 } { ö }
 }
}
```

We will avoid getting into collations orders and leave this for external sorting libraries.

```

\cxset{locale~finnish/.style = {
  locale/finnish/captions/refname        = Viitteet,
  locale/finnish/captions/abstractname   = Tiivistelmä,
```

```
  locale/finnish/captions/bibname        = Kirjallisuutta,
  locale/finnish/captions/prefacename    = Esipuhe,
  locale/finnish/captions/chaptername    = Luku,
  locale/finnish/captions/appendixname   = Liite,
  locale/finnish/captions/contentsname   = Sisältö,
  locale/finnish/captions/listfigurename = Kuvat,
  locale/finnish/captions/listtablename  = Taulukot,
  locale/finnish/captions/indexname      = Hakemisto,
  locale/finnish/captions/figurename     = Kuva,
  locale/finnish/captions/tablename      = Taulukko,
  locale/finnish/captions/partname       = Osa,
  locale/finnish/captions/pagename       = Sivu,
  locale/finnish/captions/seename        = katso,
  locale/finnish/captions/alsoname       = katso~myös,
  locale/finnish/captions/enclname       = Liitteet,
  locale/finnish/captions/ccname         = Jakelu,
  locale/finnish/captions/headtoname     = Vastaanottaja,
  locale/finnish/captions/proofname      = Todistus,
  locale/finnish/captions/glossaryname   = Sanasto,
  locale/finnish/date ={
    \number\day.\space
    \l_phd_months_wide_finnish{\month}
    \space\number\year},
}}
\cxset{locale~Finnish/.alias = locale~finnish}
```

finnish

Section A.27

Friulan

See Friulian Journal of Science.

```
\cs_set:Npn \l_phd_months_wide_friulian#1 {
  \if_case:w #1
   \or: Genâr
   \or: Fevrâr
   \or: Març
   \or: Avril
   \or: Mai
   \or: Jugn
   \or: Lui
   \or: Avost
   \or: Setembar
   \or: Otobar
   \or: Novembar
   \or: Dicembar
```

```
  \fi:
}
\cxset{locale~friulan/.style = {
  locale/friulan/captions/refname = ,
  locale/friulan/captions/abstractname   = Somari,
  locale/friulan/captions/bibname        = Bibliografie,
  locale/friulan/captions/prefacename    = Prefazion,
  locale/friulan/captions/chaptername    = Cjapitul,
  locale/friulan/captions/appendixname   = Zonte,
  locale/friulan/captions/contentsname   = Tabele~gjenerâl,
  locale/friulan/captions/listfigurename = Liste~des~figuris,
  locale/friulan/captions/listtablename  = Liste~des~tabelis,
  locale/friulan/captions/indexname      = Tabele~analitiche,
  locale/friulan/captions/figurename    = Figure,
  locale/friulan/captions/tablename      = Tabele,
  locale/friulan/captions/partname       = Part,
  locale/friulan/captions/pagename       = Pagjine,
  locale/friulan/captions/seename        = cjale,
  locale/friulan/captions/alsoname       = cjale~ancje,
  locale/friulan/captions/enclname       = Zonte(is),
  locale/friulan/captions/ccname         = Cun~copie~a,
  locale/friulan/captions/headtoname     = Par,
  locale/friulan/captions/proofname      = Dimostrazion,
  locale/friulan/captions/glossaryname   = Glossari,
  locale/friulan/date ={\number\day\space di\space
  \l_phd_months_wide_friulian {\month}
  \space dal\space\number\year},
}}
```

Section A.28

Galician

```
\cs_set:Npn \l_phd_months_wide_galician #1 {
  \if_case:w #1
   \or: Genâr
   \or: Fevrâr
   \or: Març
   \or: Avril
   \or: Mai
   \or: Jugn
   \or: Lui
   \or: Avost
   \or: Setembar
   \or: Otobar
   \or: Novembar
   \or: Dicembar
```

```
  \fi:
}
```

Section A.29

Galician

```
\cxset{locale~galician/.style = {
  locale/galician/captions/refname        = Referencias ,
  locale/galician/captions/abstractname   = Resumo,
  locale/galician/captions/bibname        = Bibliografía,
  locale/galician/captions/prefacename    = Prefacio,
  locale/galician/captions/chaptername    = Capítulo,
  locale/galician/captions/appendixname   = Apéndice,
  locale/galician/captions/contentsname   = Índice~Xeral,
  locale/galician/captions/listfigurename = Índice~de~Figuras,
  locale/galician/captions/listtablename  = Índice~de~Táboas,
  locale/galician/captions/indexname      = Índice~de~Materias,
  locale/galician/captions/figurename     = Figura,
  locale/galician/captions/tablename      = Táboa,
  locale/galician/captions/partname       = Parte,
  locale/galician/captions/pagename       = Páxina,
  locale/galician/captions/seename        = véxase,
  locale/galician/captions/alsoname       = véxase~tamén,
  locale/galician/captions/enclname       = Adxunto,
  locale/galician/captions/ccname         = Copia~a,
  locale/galician/captions/headtoname     = A,
  locale/galician/captions/proofname      = Demostración,
  locale/galician/captions/glossaryname   = Glosario,
  locale/galician/date ={\number\day~de\space\ifcase\month\or
    xaneiro\or febreiro\or marzo\or abril\or maio\or xuño\or
    xullo\or agosto\or setembro\or outubro\or novembro\or decembro\fi
    \space de~\number\year},
}}
```

galician

Section A.30

Icelandic

```
\cxset{locale~icelandic/.style = {
```

```
  locale/icelandic/captions/refname        = Heimildir,
  locale/icelandic/captions/abstractname   = Útdráttur,
  locale/icelandic/captions/bibname        = Heimildir,
  locale/icelandic/captions/prefacename    = Formáli,
  locale/icelandic/captions/chaptername    = Kafli,
  locale/icelandic/captions/appendixname   = Viðauki,
  locale/icelandic/captions/contentsname   = Efnisyfirlit ,
  locale/icelandic/captions/listfigurename = Myndaskrá,
  locale/icelandic/captions/listtablename  =Töfluskrá,
  locale/icelandic/captions/indexname      = Atriðisorðaskrá,
  locale/icelandic/captions/figurename     = Mynd,
  locale/icelandic/captions/tablename      = Tafla,
  locale/icelandic/captions/partname       = Hluti,
  locale/icelandic/captions/pagename       = Blaðsíða,
  locale/icelandic/captions/seename        = Sjá,
  locale/icelandic/captions/alsoname       = Sjá einnig,
  locale/icelandic/captions/enclname       = Hjálagt,
  locale/icelandic/captions/ccname         = Samrit,
  locale/icelandic/captions/headtoname     = Til:,
  locale/icelandic/captions/proofname      = Sönnun,
  locale/icelandic/captions/glossaryname   = Orðalisti,
  locale/icelandic/date ={\number\day.\space\ifcase\month\or
    janúar\or febrúar\or mars\or apríl\or maí\or
    júní\or júlí\or ágúst\or september\or
    október\or nóvember\or desember\fi
    \space\number\year},
}}
```

icelandic

Section A.31

Irish

```
\cxset{locale~irish/.style = {
  locale/irish/captions/refname        = Tagairtí,
  locale/irish/captions/abstractname   = Achoimre,
  locale/irish/captions/bibname        = Leabharliosta,
  locale/irish/captions/prefacename    = Réamhrá,
  locale/irish/captions/chaptername    = Tagairtí,
  locale/irish/captions/appendixname   = Aguisín ,
  locale/irish/captions/contentsname   = Clár~Ábhair,
  locale/irish/captions/listfigurename = Léaráidí ,
  locale/irish/captions/listtablename  = Táblaí ,
  locale/irish/captions/indexname      = Innéacs,
  locale/irish/captions/figurename     = Léaráid,
  locale/irish/captions/tablename      = Tábla,
  locale/irish/captions/partname       = Cuid,
  locale/irish/captions/pagename       = Leathanach,
```

```
  locale/irish/captions/seename        = féach,
  locale/irish/captions/alsoname       = féach~freisin,
  locale/irish/captions/enclname       = faoi~iamh,
  locale/irish/captions/ccname         = cc,
  locale/irish/captions/headtoname     = Go,
  locale/irish/captions/proofname      = Cruthúnas,
  locale/irish/captions/glossaryname   = Glossary,
  locale/irish/date ={\number\day\space \ifcase\month\or
    Eanáir\or Feabhra\or Márta\or Aibreán\or
    Bealtaine\or Meitheamh\or Iúil\or Lúnasa\or
    Meán Fómhair\or Deireadh Fómhair\or
    Mí na Samhna\or Mí na Nollag\fi
    \space \number\year},
}}
```

irish

Section A.32

Latin

```
\cxset{locale~latin/.style = {
  locale/latin/captions/refname = Conspectus librorum,
  locale/latin/captions/abstractname = Summarium,
  locale/latin/captions/bibname       = Conspectus librorum,
  locale/latin/captions/prefacename   = Praefatio, % change for medieval
  locale/latin/captions/chaptername   = Caput,
  locale/latin/captions/appendixname  = Additamentum,
  locale/latin/captions/contentsname = Index,
  locale/latin/captions/listfigurename = Conspectus~descriptionum,
  locale/latin/captions/listtablename = Conspectus~tabularum,
  locale/latin/captions/indexname = Index~rerum~notabilium,
  locale/latin/captions/figurename  = Descriptio,
  locale/latin/captions/tablename = Tabula,
  locale/latin/captions/partname = Pars,
  locale/latin/captions/pagename = charta,
  locale/latin/captions/seename  = cfr.,
  locale/latin/captions/alsoname = cfr.,
  locale/latin/captions/enclname = Additur,
  locale/latin/captions/ccname = Exemplar,
  locale/latin/captions/headtoname =,
  locale/latin/captions/proofname = Demonstratio,
  locale/latin/captions/glossaryname = Glossarium,
  locale/latin/date ={\uppercase\expandafter{\romannumeral\day}%
      \space \ifcase\month%
      \or Januarii\or Februarii\or Martii\or Aprilis\or Maji\or
      Junii\or Julii\or Augusti\or Septembris\or Octobris\or
         Novembris
      \or Decembris\fi%
```

```
      \space \uppercase\expandafter{\romannumeral\year}},
}}
```

Section A.33

Latvian

```
\cxset{locale~latvian/.style = {
  locale/latvian/captions/refname        = Literatūras~saraksts,
  locale/latvian/captions/abstractname   = Anotācija,
  locale/latvian/captions/bibname        = Literatūra,
  locale/latvian/captions/prefacename    = Priekšvārds,
  locale/latvian/captions/chaptername    = Nodaļa,
  locale/latvian/captions/appendixname   = Pielikums,
  locale/latvian/captions/contentsname   = Saturs,
  locale/latvian/captions/listfigurename = Attēlu~saraksts,
  locale/latvian/captions/listtablename  = Tabulu~saraksts,
  locale/latvian/captions/indexname      = Index,
  locale/latvian/captions/figurename     = Att.,
  locale/latvian/captions/tablename      = Tabula,
  locale/latvian/captions/partname       = Daļa,
  locale/latvian/captions/pagename       = lpp.,
  locale/latvian/captions/seename        = sk.,
  locale/latvian/captions/alsoname       = sk. arī,
  locale/latvian/captions/enclname       = encl,
  locale/latvian/captions/ccname         = cc,
  locale/latvian/captions/headtoname     = To,
  locale/latvian/captions/proofname      = Pierādījums,
  locale/latvian/captions/glossaryname   =,
  locale/latvian/date ={\number\year.\thinspace gada%
      \space\number\day.\thinspace%
      \ifcase\month\or%
      janvārī\or februārī\or martā\or%
      aprīlī\or maijā\or jūnijā\or%
      jūlijā\or augustā\or septembrī\or%
      oktobrī\or novembrī\or decembrī\fi},
}}
```

Section A.34

Lithuanian

```
\cxset{locale~lithuanian/.style = {
  locale/lithuanian/captions/refname        = Literatūra,
  locale/lithuanian/captions/abstractname   = Santrauka,
  locale/lithuanian/captions/bibname        = Literatūra,
  locale/lithuanian/captions/prefacename    = Pratarmė,
  locale/lithuanian/captions/chaptername    = Skyrius,
  locale/lithuanian/captions/appendixname   = Priedas,
  locale/lithuanian/captions/contentsname   = Turinys,
  locale/lithuanian/captions/listfigurename = Iliustracijų~sąrašas,
  locale/lithuanian/captions/listtablename  = Lentelių~sąrašas,
  locale/lithuanian/captions/indexname      = Rodyklė,
  locale/lithuanian/captions/figurename     = pav.,
  locale/lithuanian/captions/tablename      = lentelė,
  locale/lithuanian/captions/partname       = Dalis,
  locale/lithuanian/captions/pagename       = puslapis,
  locale/lithuanian/captions/seename        = žiūrėk,
  locale/lithuanian/captions/alsoname       = taip~pat,
  locale/lithuanian/captions/enclname       = Įdėta,
  locale/lithuanian/captions/ccname         = Kopijos,
  locale/lithuanian/captions/headtoname     = Kam,
  locale/lithuanian/captions/proofname      = Įrodymas,
  locale/lithuanian/captions/glossaryname   = Terminų~žodynas,
  locale/lithuanian/date ={\number\year\space m.\space\ifcase\month\or
      sausio\or
      vasario\or
      kovo\or
      balandžio\or
      gegužės\or
      birželio\or
      liepos\or
      rugpjūčio\or
      rugsėjo\or
      spalio\or
      lapkričio\or
      gruodžio\fi
      \space\number\day\space d.
      },
}}
```

Section A.35

Lsorbian

See also usorbian for Upper Sorbian.

```
\cxset{locale~lsorbian/.style = {
  locale/lsorbian/captions/refname        = Referency,
  locale/lsorbian/captions/abstractname   = Abstrakt,
  locale/lsorbian/captions/bibname        = Literatura,
```

```
  locale/lsorbian/captions/prefacename    = Zawod,
  locale/lsorbian/captions/chaptername    = Kapitl,
  locale/lsorbian/captions/appendixname   = Dodawki,
  locale/lsorbian/captions/contentsname   = Wopśimjeśe,
  locale/lsorbian/captions/listfigurename = Zapis~wobrazow,
  locale/lsorbian/captions/listtablename  = Zapis~tabulkow,
  locale/lsorbian/captions/indexname      = Indeks,
  locale/lsorbian/captions/figurename     = Wobraz,
  locale/lsorbian/captions/tablename      = Tabulka,
  locale/lsorbian/captions/partname       = Źěl,
  locale/lsorbian/captions/pagename       = Strona ,
  locale/lsorbian/captions/seename        = gl.,
  locale/lsorbian/captions/alsoname       = gl.~teke,
  locale/lsorbian/captions/enclname       = Pśiłoga,
  locale/lsorbian/captions/ccname         = CC,
  locale/lsorbian/captions/headtoname     = Komu,
  locale/lsorbian/captions/proofname      = Proof,
  locale/lsorbian/captions/glossaryname   = Glossary,
  locale/lsorbian/date ={
    \number\day.~\ifcase\month\or
    januara\or februara\or měrca\or apryla\or maja\or
    junija\or julija\or awgusta\or septembra\or oktobra\or
    nowembra\or decembra\fi
    \space \number\year},
}}
```

Section A.36

Magyar (Hungarian)

```
\cxset{locale~magyar/.style = {
  locale/magyar/captions/refname        = Hivatkozások ,
  locale/magyar/captions/abstractname   = Kivonat,
  locale/magyar/captions/bibname        = Irodalomjegyzék,
  locale/magyar/captions/prefacename    = Előszó,
  locale/magyar/captions/chaptername    = fejezet,
  locale/magyar/captions/appendixname   = Függelék,
  locale/magyar/captions/contentsname   = Tartalomjegyzék,
  locale/magyar/captions/listfigurename = Ábrák~jegyzéke,
  locale/magyar/captions/listtablename  = Táblázatok~jegyzéke,
  locale/magyar/captions/indexname      = Tárgymutató,
  locale/magyar/captions/figurename     = ábra,
  locale/magyar/captions/tablename      = táblázat,
  locale/magyar/captions/partname       = rész,
  locale/magyar/captions/pagename       = oldal,
  locale/magyar/captions/seename        = lásd,
  locale/magyar/captions/alsoname       = lásd~még,
```

```
  locale/magyar/captions/enclname       = Melléklet,
  locale/magyar/captions/ccname         = Körlevél-címzettek,
  locale/magyar/captions/headtoname     = Címzett,
  locale/magyar/captions/proofname      = Bizonyítás,
  locale/magyar/captions/glossaryname   = Szójegyzék,
  locale/magyar/date =
    {
      \number\year.\nobreakspace\ifcase\month\or
      január\or február\or március\or
      április\or május\or június\or
      július\or augusztus\or szeptember\or
      október\or november\or december\fi
    \space\number\day
    },
}}
\cxset{locale~Magyar/.alias = locale~magyar,
       locale~Hungarian/.alias = locale~magyar}
```

Section A.37

Marathi

```
\cxset{locale~marathi/.style = {
  locale/marathi/captions/refname        =  ,
  locale/marathi/captions/abstractname   = ,
  locale/marathi/captions/bibname        = ~~,
  locale/marathi/captions/prefacename    = ,
  locale/marathi/captions/chaptername    = ,
  locale/marathi/captions/appendixname   = ,
  locale/marathi/captions/contentsname   = ,
  locale/marathi/captions/listfigurename = ~ ,
  locale/marathi/captions/listtablename  = ~,
  locale/marathi/captions/indexname      = ,
  locale/marathi/captions/figurename     =  ,
  locale/marathi/captions/tablename      = ,
  locale/marathi/captions/partname       = ,
  locale/marathi/captions/pagename       = ,
  locale/marathi/captions/seename        = ,
  locale/marathi/captions/alsoname       = ~~ ,
  locale/marathi/captions/enclname       = ,
  locale/marathi/captions/ccname         = ..,
  locale/marathi/captions/headtoname     = ,
  locale/marathi/captions/proofname      = ,
  locale/marathi/captions/glossaryname   = ~,
  locale/marathi/date ={{\panunicode
     \number\day\space
      \ifcase\month\or
       \or
```

```
        \or
        \or
        \or
        \or
        \or
        \or
        \or
        \or
        \or
        \or
        \fi
      \space\number\year
    }},
}}
\cxset{locale~Marathi/.alias=locale~marathi}
```

Marathi

Section A.38

Nko

The script is RTL, it needs to be fixed for LuaLaTeX

```
\RequirePackage{nkonumbers}
\cxset{locale~nko/.style = {
  locale/nko/captions/refname        = ,
  locale/nko/captions/abstractname   = ,
  locale/nko/captions/bibname        = ~~,
  locale/nko/captions/prefacename    = ,
  locale/nko/captions/chaptername    = {{\textdir TRT  }},
  locale/nko/captions/appendixname   = ,
  locale/nko/captions/contentsname   = {{ \textdir TRT  }},
  locale/nko/captions/listfigurename = ~,
  locale/nko/captions/listtablename  = ~,
  locale/nko/captions/indexname      = ,
  locale/nko/captions/figurename     = ,
  locale/nko/captions/tablename      = ,
  locale/nko/captions/partname       = ,
  locale/nko/captions/pagename       = ,
  locale/nko/captions/seename        = ,
  locale/nko/captions/alsoname       = ~~,
  locale/nko/captions/enclname       = ,
  locale/nko/captions/ccname         = ..,
  locale/nko/captions/headtoname     =  ,
  locale/nko/captions/proofname      = ,
  locale/nko/captions/glossaryname   = ~,
  locale/nko/date ={{\panunicode\textdir TRT
  %FIX FOR NKO NUMBERS
```

```
    \nkonumber{\year}\space
    \ifcase\month
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or %
    \or \fi
    \space
    \space\nkonumber{\day}
    }},
}}
```

Section A.39

Norwegian (Norsk)

```
\cxset{locale~norsk/.style = {
  locale/norsk/captions/refname        = Referanser,
  locale/norsk/captions/abstractname   = Sammendrag,
  locale/norsk/captions/bibname        = Bibliografi,
  locale/norsk/captions/prefacename    = Forord,
  locale/norsk/captions/chaptername    = Kapittel,
  locale/norsk/captions/appendixname   = Tillegg,
  locale/norsk/captions/contentsname   = Innhold,
  locale/norsk/captions/listfigurename = Figurer,
  locale/norsk/captions/listtablename  = Tabeller,
  locale/norsk/captions/indexname      = Register,
  locale/norsk/captions/figurename     = Figur,
  locale/norsk/captions/tablename      = Tabell,
  locale/norsk/captions/partname       = Del,
  locale/norsk/captions/pagename       = Side,
  locale/norsk/captions/seename        = Se,
  locale/norsk/captions/alsoname       = Se~også,
  locale/norsk/captions/enclname       = Vedlegg,
  locale/norsk/captions/ccname         = Kopi~sendt,
  locale/norsk/captions/headtoname     = Til,
  locale/norsk/captions/proofname      = Bevis,
  locale/norsk/captions/glossaryname   = Ordliste,
  locale/norsk/date                    =
    {
```

```
      \number\day.~
      \ifcase\month
        \or januar
        \or februar
        \or mars
        \or april
        \or mai
        \or juni
        \or juli
        \or august
        \or september
        \or oktober
        \or november
        \or desember
     \fi
     \space\number\year
    },
}}
\cxset{locale~Norsk/.alias = locale~norsk,
       locale~Norwegian/.alias = locale~norsk}
```

Norsk

Section A.40

Piedmontese

```
\cxset{locale~piedmontese/.style = {
  locale/piedmontese/captions/refname        = Riferiment,
  locale/piedmontese/captions/abstractname   = Somari,
  locale/piedmontese/captions/bibname        = Bibliografìa,
  locale/piedmontese/captions/prefacename    = Prefassion,
  locale/piedmontese/captions/chaptername    = Kapittel,
  locale/piedmontese/captions/appendixname   = Gionta,
  locale/piedmontese/captions/contentsname   = Innhald,
  locale/piedmontese/captions/listfigurename = Lista~dle~figure ,
  locale/piedmontese/captions/listtablename  = Lista~dle~tabele,
  locale/piedmontese/captions/indexname      = Tàula~analìtica,
  locale/piedmontese/captions/figurename     = Figura,
  locale/piedmontese/captions/tablename      = Tabela,
  locale/piedmontese/captions/partname       = Part,
  locale/piedmontese/captions/pagename       = Pàgina,
  locale/piedmontese/captions/seename        = vëd,
  locale/piedmontese/captions/alsoname       = vëd~anche,
  locale/piedmontese/captions/enclname       = Gionta/e,
  locale/piedmontese/captions/ccname         = Con~còpia~a,
  locale/piedmontese/captions/headtoname     = Për,
  locale/piedmontese/captions/proofname      = Dimostrassion,
  locale/piedmontese/captions/glossaryname   = Glossari,
```

```
  locale/piedmontese/date ={
   \number\day\space\ifcase\month\or
      ëd~gené\or ëd~fevré\or ëd~mars\or d'avril\or ëd~maj\or ëd~giugn\or
      ëd~luj\or d'agost\or dë~stèmber\or d'otóber\or ëd~novèmber\or dë~dzèmber%
      \fi\space dal\space\number\year
  },
}}
```

piedmontese

Section A.41

Occitan

```
\cxset{locale~occitan/.style = {
  locale/occitan/captions/refname        = Referéncias,
  locale/occitan/captions/abstractname   = Resumit,
  locale/occitan/captions/bibname        = Bibliografia,
  locale/occitan/captions/prefacename    = Prefaci,
  locale/occitan/captions/chaptername    = Capítol,
  locale/occitan/captions/appendixname   = Annèx,
  locale/occitan/captions/contentsname   = Ensenhador,
  locale/occitan/captions/listfigurename = Taula~de~las~figuras,
  locale/occitan/captions/listtablename  = Taula~dels~tablèus,
  locale/occitan/captions/indexname      = Indèx,
  locale/occitan/captions/figurename     = Figura,
  locale/occitan/captions/tablename      = Tablèu,
  locale/occitan/captions/partname       = Partida,
  locale/occitan/captions/pagename       = Pagina,
  locale/occitan/captions/seename        = vejatz,
  locale/occitan/captions/alsoname       = vejatz~tanben,
  locale/occitan/captions/enclname       = Pèça~junta,
  locale/occitan/captions/ccname         = còpia~a,
  locale/occitan/captions/headtoname     = A,
  locale/occitan/captions/proofname      = Demostracion,
  locale/occitan/captions/glossaryname   = Glossari,
  locale/occitan/date ={
  {\def\occitanmonth{\ifcase\month\or
      de~genièr\or
      de~febrièr\or
      de~març\or
      d'abril\or
      de~mai\or
      de~junh\or
      de~julhet\or
      d'agost\or
      de~setembre\or
      d'octobre\or
      de~novembre\or
```

```
      de~decembre\fi
   }
   \def\occitanday{\ifcase\day\or
      1èr\else% primièr
      \number\day\fi% all other numbers
   }
   \occitanday\space \occitanmonth\space de~\number\year
   }},
}}
\cxset{locale~Occitan/.alias = locale~occitan}
```

Occitan

Section A.42

Polish

```
\cxset{locale~polish/.style = {
  locale/polish/captions/refname = Literatura,
  locale/polish/captions/abstractname   =Streszczenie ,
  locale/polish/captions/bibname        = Bibliografia,
  locale/polish/captions/prefacename    = Przedmowa,
  locale/polish/captions/chaptername    = Rozdział,
  locale/polish/captions/appendixname   = Dodatek,
  locale/polish/captions/contentsname   = Spis treści,
  locale/polish/captions/listfigurename = Spis rysunków,
  locale/polish/captions/listtablename  = Spis tabel,
  locale/polish/captions/indexname      = Indeks,
  locale/polish/captions/figurename     = Rysunek,
  locale/polish/captions/tablename      = Tabela,
  locale/polish/captions/partname       = Część,
  locale/polish/captions/pagename       = Strona,
  locale/polish/captions/seename        = Zobacz,
  locale/polish/captions/alsoname       = Zobacz też,
  locale/polish/captions/enclname       = Załącznik,
  locale/polish/captions/ccname         = Kopie:,
  locale/polish/captions/headtoname     = Do,
  locale/polish/captions/proofname      = Dowód ,
  locale/polish/captions/glossaryname   = Glossary, % no ranslation
  locale/polish/date ={\number\day\space\ifcase\month\or
      stycznia\or lutego\or marca\or kwietnia\or maja\or czerwca\or
      lipca\or sierpnia\or września\or października\or
      listopada\or grudnia\fi\space
      \number\year},
}}
```

Section A.43

Portuges

```
\cxset{locale~portuges/.style = {
  locale/portuges/captions/refname        = Referências,
  locale/portuges/captions/abstractname   = Resumo,
  locale/portuges/captions/bibname        = Bibliografia,
  locale/portuges/captions/prefacename    = Prefácio,
  locale/portuges/captions/chaptername    = Capítulo,
  locale/portuges/captions/appendixname   = Apêndice,
  locale/portuges/captions/contentsname   = Conteúdo,
  locale/portuges/captions/listfigurename = Lista~de~Figuras,
  locale/portuges/captions/listtablename  = Lista~de~Tabelas,
  locale/portuges/captions/indexname      = Índice,
  locale/portuges/captions/figurename     = Figura,
  locale/portuges/captions/tablename      = Tabela,
  locale/portuges/captions/partname       = Parte,
  locale/portuges/captions/pagename       = Página,
  locale/portuges/captions/seename        = ver,
  locale/portuges/captions/alsoname       = ver~também,
  locale/portuges/captions/enclname       = Anexo,
  locale/portuges/captions/ccname         = Com~cópia~a,
  locale/portuges/captions/headtoname     = Para,
  locale/portuges/captions/proofname      = Dowód,
  locale/portuges/captions/glossaryname   = Glossário,
  locale/portuges/date =
    {
      \number\day\space de\space\ifcase\month\or
      Janeiro\or Fevereiro\or Março\or Abril\or Maio\or Junho\or
      Julho\or Agosto\or Setembro\or Outubro\or Novembro\or Dezembro\fi
      \space de\space\number\year
    },
  }
}
```

Section A.44

Romanian

```
\cxset{locale~romanian/.style = {
```

```
  locale/romanian/captions/refname        = Bibliografie,
  locale/romanian/captions/abstractname   = Rezumat,
  locale/romanian/captions/bibname        = Bibliografie,
  locale/romanian/captions/prefacename    = Prefaţă,
  locale/romanian/captions/chaptername    = Capitolul,
  locale/romanian/captions/appendixname   = Anexa,
  locale/romanian/captions/contentsname   = Cuprins,
  locale/romanian/captions/listfigurename = Listă~de~figuri,
  locale/romanian/captions/listtablename  = Listă~de~tabele,
  locale/romanian/captions/indexname      = Glosar,
  locale/romanian/captions/figurename     = Figura,
  locale/romanian/captions/tablename      = Tabela,
  locale/romanian/captions/partname       = Partea,
  locale/romanian/captions/pagename       = Pagina,
  locale/romanian/captions/seename        = Vezi,
  locale/romanian/captions/alsoname       = Vezi~de~asemenea,
  locale/romanian/captions/enclname       = Anexă,
  locale/romanian/captions/ccname         = Copie,
  locale/romanian/captions/headtoname     = Pentru,
  locale/romanian/captions/proofname      = Demonstraţie ,
  locale/romanian/captions/glossaryname   = Glosar,
  locale/romanian/date ={\number\day\space\ifcase\month\or
    ianuarie\or februarie\or martie\or aprilie\or mai\or
    iunie\or iulie\or august\or septembrie\or octombrie\or
    noiembrie\or decembrie\fi
    \space \number\year},
}}
```

Section A.45

Romansh

```
\cxset{locale~romansh/.style = {
  locale/romansh/captions/refname        = Bibliografia,
  locale/romansh/captions/abstractname   = Recapitulaziun,
  locale/romansh/captions/bibname        = Index~bibliografic,
  locale/romansh/captions/prefacename    = Prefaziun,
  locale/romansh/captions/chaptername    = Chapitel,
  locale/romansh/captions/appendixname   = Appendix,
  locale/romansh/captions/contentsname   = Tavla~dal~cuntegn,
  locale/romansh/captions/listfigurename = Tavla~da~las~figuras,
  locale/romansh/captions/listtablename  = Tavla~da~las~tabellas,
  locale/romansh/captions/indexname      = Register~da~materias,
  locale/romansh/captions/figurename     = Figura,
  locale/romansh/captions/tablename      = Tabella,
  locale/romansh/captions/partname       = Part,
  locale/romansh/captions/pagename       = pagina,
  locale/romansh/captions/seename        = vesair,
```

```
  locale/romansh/captions/alsoname       = vesair~era,
  locale/romansh/captions/enclname       = Agiunta(s),
  locale/romansh/captions/ccname         = Copia~a,
  locale/romansh/captions/headtoname     = A,
  locale/romansh/captions/proofname      = Demonstraziun,
  locale/romansh/captions/glossaryname   = Glossari,
  locale/romansh/date ={{\ifcase\day\or1.\else ils\space\number\day\fi\space da\space
    \ifcase\month\or
    schaner\or favrer\or mars\or avrigl\or matg\or zercladur\or
    fanadur\or avust\or settember\or october\or november\or
    december\fi\space \number\year}},
}}
```

Section A.46

Sami Languages

Sami languages (/smi/[5]) is a group of Uralic languages spoken by the Sami people in Northern Europe (in parts of northern Finland, Norway, Sweden and extreme northwestern Russia). There are, depending on the nature and terms of division, ten or more Sami languages. Several names are used for the Sami languages: Saami, Sámi, Saame, Samic, Saamic, as well as the exonyms Lappish and Lappic. The last two, along with the term Lapp, are now often considered pejorative

```
\cxset{locale~samin/.style = {
  locale/samin/captions/refname        = Čujuhusat,
  locale/samin/captions/abstractname   = Čoahkkáigeassu,
  locale/samin/captions/bibname        = Girjjálašvuohta,
  locale/samin/captions/prefacename    = Ovdasátni,
  locale/samin/captions/chaptername    = Kapihttal,
  locale/samin/captions/appendixname   = Čuovus,
  locale/samin/captions/contentsname   = Sisdoallu,
  locale/samin/captions/listfigurename = Govvosat,
  locale/samin/captions/listtablename  = Tabeallat ,
  locale/samin/captions/indexname      = Registtar,
  locale/samin/captions/figurename     = Govus,
  locale/samin/captions/tablename      = Tabealla,
  locale/samin/captions/partname       = Oassi,
  locale/samin/captions/pagename       = Siidu,
  locale/samin/captions/seename        = geahča,
  locale/samin/captions/alsoname       = geahčageahča~maiddái ,
  locale/samin/captions/enclname       = Mielddus,
  locale/samin/captions/ccname         = Kopia sáddejuvvon,
  locale/samin/captions/headtoname     = Vuostáiváldi,
  locale/samin/captions/proofname      = Dowód ,
  locale/samin/captions/glossaryname   = Sátnelistu,
```

```
  locale/samin/date ={\ifcase\month\or
    oođđajagemánu\or
    guovvamánu\or
    njukčamánu\or
    cuoománu\or
    miessemánu\or
    geassemánu\or
    suoidnemánu\or
    borgemánu\or
    čakčamánu\or
    golggotmánu\or
    skábmamánu\or
    juovlamánu\fi
    \space\number\day.\space b.\space\number\year},
}}
```

Section A.47

Scottish

```
\cxset{locale~scottish/.style = {
  locale/scottish/captions/refname = Iomraidh,
  locale/scottish/captions/abstractname   = Brìgh ,
  locale/scottish/captions/bibname        = Leabhraichean,
  locale/scottish/captions/prefacename    = Preface, % needs translation
  locale/scottish/captions/chaptername    = Caibideil,
  locale/scottish/captions/appendixname   = Ath-sgr`ıobhadh,
  locale/scottish/captions/contentsname   = Clàr-obrach,
  locale/scottish/captions/listfigurename = Liosta~Dhealbh,
  locale/scottish/captions/listtablename  = Liosta~Chlàr,
  locale/scottish/captions/indexname      = Clàr-innse,
  locale/scottish/captions/figurename     = Dealbh ,
  locale/scottish/captions/tablename      = Clàr ,
  locale/scottish/captions/partname       = Cuid,
  locale/scottish/captions/pagename       = t.d.,
  locale/scottish/captions/seename        = see, % needs translation
  locale/scottish/captions/alsoname       = ,
  locale/scottish/captions/enclname       = a-staigh,
  locale/scottish/captions/ccname         = lethbhreac~gu,
  locale/scottish/captions/headtoname     = gu,
  locale/scottish/captions/proofname      = Proof,
  locale/scottish/captions/glossaryname   = Glossary,
  locale/scottish/date ={\number\day\space \ifcase\month\or
    am~Faoilteach\or an~Gearran\or am~Màrt\or an~Giblean\or
    an~Cèitean\or an~t-Òg mhios\or an~t-Iuchar\or
    Lùnasdal\or an~Sultuine\or an~Dàmhar\or
    an~t-Samhainn\or an~Dubhlachd\fi
```

```
    \space \number\year},
}}
```

Section A.48

Serbian

Standard Serbian language uses both Cyrillic (ћирилица, ćirilica) and Latin script (latinica, латиница). Serbian is a rare example of synchronic digraphia, a situation where all literate members of a society have two interchangeable writing systems available to them. Media and publishers typically select one alphabet or another. The default is for latin script.

```
\cxset{locale~serbian/.style = {
  locale/serbian/captions/refname        = Bibliografija,
  locale/serbian/captions/abstractname   = Sažetak,
  locale/serbian/captions/bibname        = Literatura,
  locale/serbian/captions/prefacename    = Predgovor,
  locale/serbian/captions/chaptername    = Glava,
  locale/serbian/captions/appendixname   = Dodatak,
  locale/serbian/captions/contentsname   = Sadržaj,
  locale/serbian/captions/listfigurename = Spisak slika,
  locale/serbian/captions/listtablename  = Spisak tabela,
  locale/serbian/captions/indexname      = Registar,
  locale/serbian/captions/figurename     = Slika,
  locale/serbian/captions/tablename      = Tabela ,
  locale/serbian/captions/partname       = Deo,
  locale/serbian/captions/pagename       = Strana ,
  locale/serbian/captions/seename        = Vidi,
  locale/serbian/captions/alsoname       = Vidi takođe ,
  locale/serbian/captions/enclname       = Prilozi,
  locale/serbian/captions/ccname         = Kopije,
  locale/serbian/captions/headtoname     = Prima,
  locale/serbian/captions/proofname      = Dokaz,
  locale/serbian/captions/glossaryname   = Rečnik~nepoznatih~reči,
  locale/serbian/date ={\number\day .\space\ifcase\month\or
    januar\or februar\or mart\or april\or maj\or
    jun\or jul\or avgust\or septembar\or oktobar\or novembar\or
    decembar\fi \space \number\year.},
}}
\cxset{locale~serbian~cyrillic/.style = {
  locale/serbian~cyrillic/captions/refname        = ,
  locale/serbian~cyrillic/captions/abstractname   = ,
  locale/serbian~cyrillic/captions/bibname        = ,
  locale/serbian~cyrillic/captions/prefacename    = ,
  locale/serbian~cyrillic/captions/chaptername    = ,
  locale/serbian~cyrillic/captions/appendixname   = ,
  locale/serbian~cyrillic/captions/contentsname   = ,
```

```
  locale/serbian~cyrillic/captions/listfigurename = ~,
  locale/serbian~cyrillic/captions/listtablename  =  ,
  locale/serbian~cyrillic/captions/indexname      = ,
  locale/serbian~cyrillic/captions/figurename     = ,
  locale/serbian~cyrillic/captions/tablename      = ,
  locale/serbian~cyrillic/captions/partname       = ,
  locale/serbian~cyrillic/captions/pagename       = ,
  locale/serbian~cyrillic/captions/seename        = ,
  locale/serbian~cyrillic/captions/alsoname       =  ,
  locale/serbian~cyrillic/captions/enclname       = ,
  locale/serbian~cyrillic/captions/ccname         = ,
  locale/serbian~cyrillic/captions/headtoname     = ,
  locale/serbian~cyrillic/captions/proofname      = ,
  locale/serbian~cyrillic/captions/glossaryname   =   ,
  locale/serbian~cyrillic/date ={\number\day .\space\ifcase\month\or
    \or \or \or \or \or
    \or \or \or \or \or \or
    \fi \space \number\year.},
}}
```

Section A.49

Slovak

```
\cxset{locale~slovak/.style = {
  locale/slovak/captions/refname        = Referencie,
  locale/slovak/captions/abstractname   = Abstrakt,
  locale/slovak/captions/bibname        = Literatúra,
  locale/slovak/captions/prefacename    = Úvod,
  locale/slovak/captions/chaptername    = Kapitola,
  locale/slovak/captions/appendixname   = Dodatok,
  locale/slovak/captions/contentsname   = Obsah,
  locale/slovak/captions/listfigurename = Zoznam~obrázkov ,
  locale/slovak/captions/listtablename  = Zoznam~tabuliek,
  locale/slovak/captions/indexname      = Index,
  locale/slovak/captions/figurename     = Obrázok,
  locale/slovak/captions/tablename      = Tabuľka,
  locale/slovak/captions/partname       = Časť,
  locale/slovak/captions/pagename       = Strana,
  locale/slovak/captions/seename        = viď,
  locale/slovak/captions/alsoname       = viď~tiež,
  locale/slovak/captions/enclname       = Prílohy,
  locale/slovak/captions/ccname         = cc.,
  locale/slovak/captions/headtoname     = Pre,
  locale/slovak/captions/proofname      = Dôkaz,
  locale/slovak/captions/glossaryname   = Slovník,
  locale/slovak/date ={\number\day.~\ifcase\month\or
```

```
    januára\or februára\or marca\or apríla\or mája\or
    júna\or júla\or augusta\or septembra\or októbra\or
    novembra\or decembra\fi
    \space \number\year},
}}
```

Section A.50

Slovenian

```
\cxset{locale~slovenian/.style = {
  locale/slovenian/captions/refname        = Literatura,
  locale/slovenian/captions/abstractname   = Povzetek,
  locale/slovenian/captions/bibname        = Literatura,
  locale/slovenian/captions/prefacename    = Predgovor,
  locale/slovenian/captions/chaptername    = Poglavje,
  locale/slovenian/captions/appendixname   = Dodatek,
  locale/slovenian/captions/contentsname   = Kazalo,
  locale/slovenian/captions/listfigurename = Slike,
  locale/slovenian/captions/listtablename  = Tabele,
  locale/slovenian/captions/indexname      = Stvarno~kazalo,
  locale/slovenian/captions/figurename     = Slika,
  locale/slovenian/captions/tablename      = Tabela,
  locale/slovenian/captions/partname       = Del,
  locale/slovenian/captions/pagename       = Stran,
  locale/slovenian/captions/seename        = glej,
  locale/slovenian/captions/alsoname       = glej~tudi,
  locale/slovenian/captions/enclname       = Priloge,
  locale/slovenian/captions/ccname         = Kopije,
  locale/slovenian/captions/headtoname     = Prejme,
  locale/slovenian/captions/proofname      = Dokaz,
  locale/slovenian/captions/glossaryname   = Slovar,
  locale/slovenian/date ={\number\day.~\ifcase\month\or
    januar\or februar\or marec\or april\or maj\or junij\or
    julij\or avgust\or september\or oktober\or november\or december\fi
    \space \number\year},
}}
% Alphabet consists of 25 lower and 25 upper letters
\cs_new:Npn \int_to_Alph_slovenian:n #1
 {
 \int_to_symbols:nnn {#1} { 25 }
 {
  { 1 } { A }
  { 2 } { B }
  { 3 } { C }
  { 4 } { Č }
  { 5 } { D }
  { 6 } { E }
```

```
  { 7 } { F }
  { 8 } { G }
  { 9 } { H }
  { 10 } { I }
  { 11 } { J }
  { 12 } { K }
  { 13 } { L }
  { 14 } { M }
  { 15 } { N }
  { 16 } { O }
  { 17 } { P }
  { 18 } { R }
  { 19 } { S }
  { 20 } { Š }
  { 21 } { T }
  { 22 } { U }
  { 23 } { V }
  { 24 } { Z }
  { 25 } { Ž }
 }
}

\cs_new:Npn \int_to_alph_slovenian:n #1
 {
 \int_to_symbols:nnn {#1} { 25 }
 {
  { 1 } { a }
  { 2 } { b }
  { 3 } { c }
  { 4 } { č }
  { 5 } { d }
  { 6 } { e }
  { 7 } { f }
  { 8 } { g }
  { 9 } { h }
  { 10 } { i }
  { 11 } { j }
  { 12 } { k }
  { 13 } { l }
  { 14 } { m }
  { 15 } { n }
  { 16 } { o }
  { 17 } { p }
  { 18 } { r }
  { 19 } { s }
  { 20 } { š }
  { 21 } { t }
  { 22 } { u }
  { 23 } { v }
  { 24 } { z }
  { 25 } { ž }
 }
}
```

Section A.51

Spanish

```
\cxset{locale~spanish/.style = {
  locale/spanish/captions/refname        = Referencias,
  locale/spanish/captions/abstractname   = Resumen,
  locale/spanish/captions/bibname        = Bibliografía,
  locale/spanish/captions/prefacename    = Prefacio,
  locale/spanish/captions/chaptername    = Capítulo,
  locale/spanish/captions/appendixname   = Apéndice,
  locale/spanish/captions/contentsname   = Índice~general,
  locale/spanish/captions/listfigurename = Índice~de~figuras,
  locale/spanish/captions/listtablename  = Índice~de~cuadros,
  locale/spanish/captions/indexname      = Índice~alfabético,
  locale/spanish/captions/figurename     = Figura,
  locale/spanish/captions/tablename      = Cuadro,
  locale/spanish/captions/partname       = Parte,
  locale/spanish/captions/pagename       = Página,
  locale/spanish/captions/seename        = véase,
  locale/spanish/captions/alsoname       = véase~también,
  locale/spanish/captions/enclname       = Adjunto(s),
  locale/spanish/captions/ccname         = Copia~a,
  locale/spanish/captions/headtoname     = A,
  locale/spanish/captions/proofname      = Prueba,
  locale/spanish/captions/glossaryname   = Glosario,
  locale/spanish/date ={\number\day~de~\ifcase\month\or
    enero\or febrero\or marzo\or abril\or mayo\or junio\or
    julio\or agosto\or septiembre\or octubre\or noviembre\or
    diciembre\fi\space de~\number\year},
}}
```

Section A.52

Swedish

```
\cxset{locale~swedish/.style = {
  locale/swedish/captions/refname        = Referenser,
  locale/swedish/captions/abstractname   = Sammanfattning,
  locale/swedish/captions/bibname        = Litteraturförteckning,
  locale/swedish/captions/prefacename    = Förord,
```

```
  locale/swedish/captions/chaptername    = Kapitel,
  locale/swedish/captions/appendixname   = Bilaga,
  locale/swedish/captions/contentsname   = Innehåll,
  locale/swedish/captions/listfigurename = Figurer,
  locale/swedish/captions/listtablename  = Tabeller,
  locale/swedish/captions/indexname      = Sakregister,
  locale/swedish/captions/figurename     = Figur,
  locale/swedish/captions/tablename      = Tabell,
  locale/swedish/captions/partname       = Del,
  locale/swedish/captions/pagename       = Sida,
  locale/swedish/captions/seename        = se,
  locale/swedish/captions/alsoname       = se~även,
  locale/swedish/captions/enclname       = Bil.,
  locale/swedish/captions/ccname         = Kopia~för~kännedom,
  locale/swedish/captions/headtoname     = Till,
  locale/swedish/captions/proofname      = Bevis,
  locale/swedish/captions/glossaryname   = Ordlista,
  locale/swedish/date ={\number\day\space\ifcase\month\or
    januari\or februari\or mars\or april\or maj\or juni\or
    juli\or augusti\or september\or oktober\or november\or
    december\fi
    \space\number\year},
}}
```

swedish

Section A.53

Tamil

```
\cxset{locale~tamil/.style = {
  locale/tamil/captions/refname = ,
  locale/tamil/captions/abstractname = ,
  locale/tamil/captions/bibname       = ,
  locale/tamil/captions/prefacename   = ,
  locale/tamil/captions/chaptername   = ,
  locale/tamil/captions/appendixname  = ,
  locale/tamil/captions/contentsname  = ,
  locale/tamil/captions/listfigurename = ~,
  locale/tamil/captions/listtablename = ~,
  locale/tamil/captions/indexname = ,
  locale/tamil/captions/figurename  = ,
  locale/tamil/captions/tablename = ,
  locale/tamil/captions/partname = ,
  locale/tamil/captions/pagename = ,
  locale/tamil/captions/seename  = ,
  locale/tamil/captions/alsoname = ,
  locale/tamil/captions/enclname = {},
  locale/tamil/captions/ccname = {},
```

```
  locale/tamil/captions/headtoname ={} ,
  locale/tamil/captions/proofname = ,
  locale/tamil/captions/glossaryname = ,
  locale/tamil/date ={\number\year\space\ifcase\month\or
     \or
     \or
     \or
    \or
     \or
     \or
     \or
    \or
     \or
     \or
     \or
     \fi
     \space\number\day},
}}
```

tamil

Section A.54

Telugu

```
\cxset{locale~telugu/.style = {
  locale/telugu/captions/refname = ,
  locale/telugu/captions/abstractname = ,
  locale/telugu/captions/bibname       =  ,
  locale/telugu/captions/prefacename   = ,
  locale/telugu/captions/chaptername   = ,
  locale/telugu/captions/appendixname  = ,
  locale/telugu/captions/contentsname  = ,
  locale/telugu/captions/listfigurename =  ,
  locale/telugu/captions/listtablename =  ,
  locale/telugu/captions/indexname =  ,
  locale/telugu/captions/figurename  = ,
  locale/telugu/captions/tablename = ,
  locale/telugu/captions/partname = ,
  locale/telugu/captions/pagename = ,
  locale/telugu/captions/seename  = ,
  locale/telugu/captions/alsoname =  ,
  locale/telugu/captions/enclname = *,
  locale/telugu/captions/ccname = ,
  locale/telugu/captions/headtoname = ,
  locale/telugu/captions/proofname = ,
  locale/telugu/captions/glossaryname = ,
  locale/telugu/date ={{\panunicode
```

```
  \def\telugu@month{
    \ifcase\month\or
         \or
         \or
         \or
         \or
         \or
         \or
         \or
         \or
         \or
         \or
         \or
         \fi}
   \telugu@month\space\number\day,\space\number\year}},
}}
```

telugu

Section A.55

Turkish

```
\cxset{locale~turkish/.style = {
  locale/turkish/captions/refname        = Kaynaklar,
  locale/turkish/captions/abstractname   = Özet,
  locale/turkish/captions/bibname        = Kaynakça,
  locale/turkish/captions/prefacename    = Önsöz,
  locale/turkish/captions/chaptername    = Bölüm,
  locale/turkish/captions/appendixname   = Ek,
  locale/turkish/captions/contentsname   = İçindekiler,
  locale/turkish/captions/listfigurename = Şekil Listesi,
  locale/turkish/captions/listtablename  = Tablo Listesi,
  locale/turkish/captions/indexname      = Dizin,
  locale/turkish/captions/figurename     = Şekil,
  locale/turkish/captions/tablename      = Tablo,
  locale/turkish/captions/partname       = Kısım,
  locale/turkish/captions/pagename       = Sayfa,
  locale/turkish/captions/seename        = bkz.,
  locale/turkish/captions/alsoname       = ayrıca~bkz.,
  locale/turkish/captions/enclname       = İlişik,
  locale/turkish/captions/ccname         = Diğer~Alıcılar,
  locale/turkish/captions/headtoname     = Alıcı,
  locale/turkish/captions/proofname      = Kanıt,
  locale/turkish/captions/glossaryname   = Lügatçe,
  locale/turkish/date ={\number\day\space\ifcase\month\or
    Ocak\or Şubat\or Mart\or Nisan\or Mayıs\or Haziran\or
    Temmuz\or Ağustos\or Eylül\or Ekim\or Kasım\or
    Aralık\fi
```

```
    \space\number\year},
}}
```

Section A.56

Turkmen

```
\cxset{locale~turkmen/.style = {
  locale/turkmen/captions/refname        = Çeşmeler,
  locale/turkmen/captions/abstractname   = Gysgaça~manysy,
  locale/turkmen/captions/bibname        = Çeşmeler,
  locale/turkmen/captions/prefacename    = Sözbaşy,
  locale/turkmen/captions/chaptername    = Bap,
  locale/turkmen/captions/appendixname   = Goşmaça,
  locale/turkmen/captions/contentsname   = Mazmuny,
  locale/turkmen/captions/listfigurename = Suratlaryň~sanawy,
  locale/turkmen/captions/listtablename  = Tablisalaryň~sanawy,
  locale/turkmen/captions/indexname      = Indeks,
  locale/turkmen/captions/figurename     = Surat,
  locale/turkmen/captions/tablename      = Tablisa,
  locale/turkmen/captions/partname       = Bölüm,
  locale/turkmen/captions/pagename       = Sahypa,
  locale/turkmen/captions/seename        = ser.,
  locale/turkmen/captions/alsoname       = şuňa-da~ser.,
  locale/turkmen/captions/enclname       = Goşmaça,
  locale/turkmen/captions/ccname         = Iberilenler,
  locale/turkmen/captions/headtoname     = Kime,
  locale/turkmen/captions/proofname      = Delil,
  locale/turkmen/captions/glossaryname   = Sözlük,
  locale/turkmen/date = {\number\day\space\ifcase\month\or
    Ýanwar\or Fewral\or Mart\or Aprel\or Maý\or Iýun\or
    Iýul\or Awgust\or Sentýabr\or Oktýabr\or Noýabr\or
    Dekabr\fi
    \space\number\year},
}}
```

turkmen

Section A.57

Ukrainian

Ukrainian uk uk 1058 422 1251

```
\cxset{locale~ukrainian/.style = {
  locale/ukrainian/captions/refname        = ,
  locale/ukrainian/captions/abstractname   = ,
  locale/ukrainian/captions/bibname        = ,
  locale/ukrainian/captions/prefacename    = ,
  locale/ukrainian/captions/chaptername    = ,
  locale/ukrainian/captions/appendixname   = ,
  locale/ukrainian/captions/contentsname   = ,
  locale/ukrainian/captions/listfigurename = ~,
  locale/ukrainian/captions/listtablename  = ~,
  locale/ukrainian/captions/indexname      = ,
  locale/ukrainian/captions/figurename     = .,
  locale/ukrainian/captions/tablename      = .,
  locale/ukrainian/captions/partname       = ,
  locale/ukrainian/captions/pagename       = .,
  locale/ukrainian/captions/seename        = .,
  locale/ukrainian/captions/alsoname       = .\ ~,
  locale/ukrainian/captions/enclname       = ,
  locale/ukrainian/captions/ccname         = ,
  locale/ukrainian/captions/headtoname     = ,
  locale/ukrainian/captions/proofname      = ,
  locale/ukrainian/captions/glossaryname   = ~,
  locale/ukrainian/date ={\number\day\space\ifcase\month\or
    \or
    \or
    \or
    \or
    \or
    \or
    \or
    \or
    \or
    \or
    \or
    \fi
    \space\number\year\space .},
}}
```

Section A.58

Usorbian

The Sorbian languages (Upper Sorbian: Serbska rěč, Lower Sorbian: Serbska rěc) are two closely related, but only partially mutually intelligible, West Slavic languages spoken by the Sorbs, a West Slavic minority in the Lusatia region of

eastern Germany. They are classified under the West Slavic branch of the Indo-European languages and are therefore closely related to the other two West Slavic subgroups: Lechitic and Czech–Slovak.[2] Historically the languages have also been known as Wendish (named after the Wends, earliest Slavic people in modern Poland and Germany) or Lusatian. Their collective ISO 639-2 code is `wen`.

There are two literary languages: Upper Sorbian (hornjoserbsce), spoken by about 40,000 people in Saxony, and Lower Sorbian (dolnoserbski) spoken by about 10,000 people in Brandenburg. The area where the two languages are spoken is known as Lusatia (Łužica in Upper Sorbian, Łužyca in Lower Sorbian, or Lausitz in German).

```
\cxset{locale~usorbian/.style = {
  locale/usorbian/captions/refname        = Referency ,
  locale/usorbian/captions/abstractname   = Abstrakt,
  locale/usorbian/captions/bibname        = Literatura ,
  locale/usorbian/captions/prefacename    = Zawod,
  locale/usorbian/captions/chaptername    = Kapitl,
  locale/usorbian/captions/appendixname   = Dodawki,
  locale/usorbian/captions/contentsname   = Wobsah,
  locale/usorbian/captions/listfigurename = Zapis~wobrazow,
  locale/usorbian/captions/listtablename  = Zapis~tabulkow,
  locale/usorbian/captions/indexname      = Indeks,
  locale/usorbian/captions/figurename     = Wobraz,
  locale/usorbian/captions/tablename      = Tabulka,
  locale/usorbian/captions/partname       = Dźěl,
  locale/usorbian/captions/pagename       = Strona,
  locale/usorbian/captions/seename        = hl.,
  locale/usorbian/captions/alsoname       = hl.~tež,
  locale/usorbian/captions/enclname       = Přłoha,
  locale/usorbian/captions/ccname         = CC,
  locale/usorbian/captions/headtoname     = Komu ,
  locale/usorbian/captions/proofname      = Proof,
  locale/usorbian/captions/glossaryname   = Glossary,
  locale/usorbian/date                    =
    {
      \number\day.\space
      \ifcase\month
        \or januara
        \or februara
        \or měrca
        \or apryla
        \or meje
        \or junija
        \or julija
        \or awgusta
        \or septembra
        \or oktobra
        \or nowembra
        \or decembra
```

[2] Online article by Hermut Feska in: About Sorbian Language.

```
      \fi
      \space\number\year
    },
}}
\cxset{locale~Usorbian/.alias = locale~usorbian}
```

Usorbian

Section A.59

hangul

```

\cxset{locale~hangul/.style = {
  locale/hangul/captions/refname        = Tài~liệu,
  locale/hangul/captions/abstractname   = Tóm~tắt~nội~dung,
  locale/hangul/captions/bibname        = Tài~liệu~tham~khảo,
  locale/hangul/captions/prefacename    = Lời~nói~đầu,
  locale/hangul/captions/chaptername    = Chương,
  locale/hangul/captions/appendixname   = Phụ~lục,
  locale/hangul/captions/contentsname   = Mục~lục,
  locale/hangul/captions/listfigurename = Danh~sách~hình~vẽ ,
  locale/hangul/captions/listtablename  = Danh~sách~bảng,
  locale/hangul/captions/indexname      = Chỉ~mục,
  locale/hangul/captions/figurename     = Hình,
  locale/hangul/captions/tablename      = Bảng,
  locale/hangul/captions/partname       = Phần ,
  locale/hangul/captions/pagename       = Trang,
  locale/hangul/captions/seename        = Xem ,
  locale/hangul/captions/alsoname       = Xem~thêm,
  locale/hangul/captions/enclname       = Kèm~theo,
  locale/hangul/captions/ccname         = Cùng~gửi,
  locale/hangul/captions/headtoname     = Gửi,
  locale/hangul/captions/proofname      = Chứng~minh,
  locale/hangul/captions/glossaryname   = Từ~điển~chú~giải,
  locale/hangul/date ={Ngày\space\number\day\space
    tháng\space\number\month\space
    năm\space\number\year},
}}
\cxset{locale~hangul/.alias = locale~hangul}
```

hangul

Section A.60

Welsh

for dates and times see swansea

```
\newif\ifwelsh@first

\def\welsh@article#1{
  \welsh@firsttrue
  y
  \expandafter\welsh@article@do#1
 }

\def\welsh@article@do#1{
  \ifwelsh@first\welsh@isvowel#1
  \ifwelsh@vowel
     r\space
     \welsh@vowelfalse
   \else
   \space
   \fi#1
   \welsh@firstfalse
   \fi
 }

\newif\ifwelsh@vowel
\def\welsh@isvowel#1{
   \ifx#1a\welsh@voweltrue
   \else
   \ifx#1u
    \welsh@voweltrue
    \else
    \ifx#1w\welsh@voweltrue
    \fi\fi\fi}% FIXME Add the other vowels, just for good measure

\def\welsh@ordinal@long#1{%
  \if_case:w #1
    \or: cyntaf
    \or: ail
    \or: trydydd
    \or: pedwerydd
    \or: pumed
    \or: chweched
    \or: seithfed
    \or: wythfed
    \or: nawfed
```

```
    \or: degfed
    \or: unfed~ar~ddeg
    \or: deuddegfed
    \or: trydydd~ar~ddeg
    \or: pedwerydd~ar~ddeg
    \or: pymthegfed
    \or: unfed~ar~bymtheg
    \or: ail~ar~bymtheg
    \or: deunawfed
    \or: pedwerydd~ar~bymtheg
    \or: ugeinfed
    \else:
     \exp_after:wN \welsh@ordinalplusxx@long#1
    \fi:
  }

\def\welsh@ordinalplusxx@long#1{%
  \let\dday=#1\advance\dday~by~-20\relax\welsh@ordinal@long\dday\space ar~hugain%
}

\cxset{welsh~date~format/.is~choice,
       welsh~date~format/standard/.code =
       \def\today{\expandafter\welsh@article\welsh@ordinal@long\day\space o\space
       \if_case:w \month
       \or: Ionawr
       \or: Chwefror
       \or: Fawrth
       \or: Ebrill
       \or: Fai
       \or: Fehefin
       \or: Orffenaf
       \or: Awst
       \or: Fedi
       \or: Hydref
       \or: Dachwedd
       \or: Ragfyr
      \fi:
      \space\number\year},
     welsh~date~format/formal/.code    = \def\today{\ifcase\day\or 1af\or 2ail\or 3ydd\o
    \or 7fed\or 8fed\or 9fed\or 10fed\or 11eg\or 12fed\or 13eg\or
    14eg\or 15fed\or 16eg\or 17eg\or 18fed\or 19eg\or
    20fed\else\number\day ain\fi\space\ifcase\month\or
    Ionawr\or Chwefror\or Mawrth\or Ebrill\or
    Mai\or Mehefin\or Gorffennaf\or Awst\or
    Medi\or Hydref\or Tachwedd\or Rhagfyr\fi%
    \space\number\year},
    welsh~date~format=formal,
  }
\cxset{locale~welsh/.style = {
  locale/welsh/captions/refname        = Cyfeiriadau,
  locale/welsh/captions/abstractname   = Crynodeb,
  locale/welsh/captions/bibname        = Llyfryddiaeth,
  locale/welsh/captions/prefacename    = Rhagair,
  locale/welsh/captions/chaptername    = Pennod,
```

```
  locale/welsh/captions/appendixname   = Atodiad,
  locale/welsh/captions/contentsname   = Cynnwys,
  locale/welsh/captions/listfigurename = Rhestr~Ddarluniau,
  locale/welsh/captions/listtablename  = Rhestr~Dablau,
  locale/welsh/captions/indexname      = Mynegai,
  locale/welsh/captions/figurename     = Darlun,
  locale/welsh/captions/tablename      = Taflen,
  locale/welsh/captions/partname       = Rhan,
  locale/welsh/captions/pagename       = tudalen,
  locale/welsh/captions/seename        = gweler,
  locale/welsh/captions/alsoname       = gweler~hefyd,
  locale/welsh/captions/enclname       = amgaeëdig,
  locale/welsh/captions/ccname         = copïau,
  locale/welsh/captions/headtoname     = At,
  locale/welsh/captions/proofname      = Prawf,
  locale/welsh/captions/glossaryname   = Rhestr~termau,
  welsh~date~format=standard,
}}
\cxset{locale~Welsh/.alias=locale~welsh}
```

Section A.61

Hangul

```
\cxset{locale~hangul/.style = {
  locale/hangul/captions/refname        = ,
  locale/hangul/captions/abstractname   = ,
  locale/hangul/captions/bibname        = ,
  locale/hangul/captions/prefacename    = ,
  locale/hangul/captions/chaptername    = ,
  locale/hangul/captions/appendixname   = ,
  locale/hangul/captions/contentsname   = ~,
  locale/hangul/captions/listfigurename = ~,
  locale/hangul/captions/listtablename  = ~,
  locale/hangul/captions/indexname      = ,
  locale/hangul/captions/figurename     = ,
  locale/hangul/captions/tablename      = ,
  locale/hangul/captions/partname       = ,
  locale/hangul/captions/pagename       = ,
  locale/hangul/captions/seename        = ,
  locale/hangul/captions/alsoname       = ,
  locale/hangul/captions/enclname       = ,
  locale/hangul/captions/ccname         = ,
  locale/hangul/captions/headtoname     = ,
  locale/hangul/captions/proofname      = ,
  locale/hangul/captions/glossaryname   = ,
  locale/hangul/date ={Ngày\space\number\day\space
```

```
    tháng\space\number\month\space
    năm\space\number\year},
}}
```

hangul

Welsh The Captions List is helpful for presenting the captions in various languages. Inspired from MonTeX!!!

```
\newcommand{\CaptionsList}[1]{%
   \bgroup
   \cxset{locale~#1}
   %\ttfamily
   \panunicode%
\begin{longtable}[]{lll}
\caption{Captions~in~#1\label{#1captions}}\\
\toprule
   Command                & English        & #1 \\
 \midrule
 \cs{prefacename}       & Preface        & \prefacename\\
 \cs{refname}         & References       & \refname\\
 \cs{abstractname}     & Abstract        & \abstractname\\
 \cs{bibname}         & Bibliography     & \bibname\\
 \cs{chaptername}       & Chapter        & \chaptername\\
 \cs{appendixname}     & Appendix        & \appendixname\\
 \cs{contentsname}     & Contents        & \contentsname\\
 \cs{listfigurename}    & List~of~Figures & \listfigurename\\
 \cs{listtablename}     & List~of~Tables  & \listtablename\\
 \cs{indexname}         & Index           & \indexname\\
 \cs{figurename}       & Figure          & \figurename\\
 \cs{tablename}         & Table         & \tablename\\
 \cs{partname}         & Part           & \partname\\
 \cs{enclname}         & encl           & \enclname\\
 \cs{ccname}         & cc             & \ccname\\
 \cs{headtoname}       & To             & \headtoname\\
 \cs{pagename}         & Page           & \pagename\\
 \cs{seename}         & see            & \seename\\
   \cs{alsoname}         & see~also        & \alsoname\\
 \cs{glossaryname}      & Glossary        & \glossaryname\\
 \cs{today}             &                 & \today \\
 %\cs{dateitalian}      &                 & \dateitalian\\
 \bottomrule
 \end{longtable}
%\captionsenglish
\egroup
}

\clist_set:cn {en_clist} {, January, February, March, April, May,
   June, July, August, September, October, November, December}

\ExplSyntaxOff
```

Index

Symbols

\#(#) a772
\# 2002, 2544
\$($) a772
\%(%) a772
\% 466, 654, 1780, 2002, 2062, 2882, 2884, 2887, 2888, 2900, 2921
\&(&) a772
\(. 267
\) 267
* 267
* 1973, 1993, 2002
\- 35
\- 2002, 2526
.tex (file extension) a785
.tfm (file extension) a208
/errors/boolean expected 670
/errors/unknown choice value . . 670
/errors/unknown key 670
/errors/value forbidden 669
/errors/value required 669
/utils/exec 669
\: 267
\; 266
\@ 88
@@ commands:
\@@_doc_aux_env: 1463, 1477
\@@_doc_aux_env_star 1469, 1476
\@@_doc_aux_key: 1501, 1523
\@@_doc_aux_key_star 1512, 1522
\@@_doc_color_aux:n . 138, 153
\@@_doc_color_star:n 144, 152
\g_@@_macrocode_colorize_bool 8
\@@_print_docfile:n . 1705, 1715
\@@_document_types_clist . 881
\@@fileswith@pti@ns 519
\@@hyph 34
\@@italiccorr 35
\@@par 34
\@@unprocessedoptions 520
\@@warning 70
\@Alph 149
\@Esphack 83
\@M 80
\@Roman 149
\@activechar@info 453
\@addmarginpar 469
\@addtobot 439, 461
\@addtocurcol 439
\@addtodblcol 440
\@addtofilelist 102
\@addtonextcol 439, 466
\@addtoreset 148
\@addtotoporbot 439, 462
\@afterheading 909
\@alph 149
\@arabic 149
\@argdef 40
\@autoerr 71
\@autoload 55
\@backslashchar 49
\@badcrerr 73
\@badend 72
\@badlinearg 73
\@badmath 72
\@badpoptabs 72
\@badrequireerror 521
\@badtab 72
\@begindocumenthook 519
\@begindvi 456
\@beginparpenalty 288
\@bibitem 404
\@biblabel 406
\@bitor 437
\@boxfpsbit 474
\@break@tfor 62
\@bsphack 82, 83
\@car 39
\@cdr 39
\@cflb 457
\@changed@cmd 112
\@changed@x 112

\@charlb 100
\@charrb 100
\@check@c 47
\@check@eq 48
\@cite 406
\@cite@ofmt 406
\@citex 404
\@classoptionslist 508
\@cls@pkg 520
\@clsextension 508
\@clubpenalty 94
\@combinedblfloats 458
\@combinefloats 435, 456
\@comdblflelt 458
\@comflelt 457
\@cons 38
\@ctrerr 71
\@currentlabel 250
\@currenvir 253
\@currext 508
\@currname 508
\@currnamestack 508
\@dblarg 54
\@dblflset 372
\@dblfpbot 483
\@dblfpsep 483
\@dblfptop 483
\@dec@text@cmd 112
\@declaredoptions 507
\@definecounter 148
\@depth 35
\@dischyph 35
\@doclearpage 448
\@documentclasshook 507
\@doendpe 294
\@dofilelist 103
\@donoparitem 295
\@eha 70
\@ehb 70
\@ehc 70
\@ehd 70
\@empty 61
\@emptycol 445
\@enddocumenthook 519
\@endfloatbox 376
\@endparenv 294
\@endparpenalty 288
\@endpelttrue 295
\@ensuredmath 272
\@enumdepth 299
\@eqcnt 270
\@eqncr 271
\@eqnnum 269, 272
\@eqnsel 270
\@eqpen 270
\@esphack 83
\@expandtwoargs 48
\@filelist 102
\@fileswithoptions 516
\@firstofone 48
\@firstoftwo 48
\@flcheckspace 476
\@float 372
\@flsetnum 475
\@flsettextmin 476
\@flstop 473
\@fltovf 73
\@flupdates 477
\@fnsymbol 150
\@footnotetext 383
\@for 61
\@forloop 61
\@fornoop 61
\@fpbot 482
\@fpsep 482
\@fpstype 473
\@fptop 482
\@getfpsbit 474
\@getpen 81
\@gnewline 81
\@gobble 48
\@gobblefour 48
\@gobbletwo 48
\@height 35
\@hspace 88
\@hspacer 88
\@iden 48
\@if 46
\@ifclasslater 509
\@ifclassloaded 509
\@ifclasswith 510
\@ifdefinable 43
\@iffileonpath 100
\@ifnch 53
\@ifnextchar 52
\@iforloop 61
\@ifpackagelater 509
\@ifpackageloaded 509
\@ifpackagewith 510
\@ifstar 54
\@ifundefined 52
\@iinput 101
\@include 98
\@inlabel 288
\@inmatherr 74
\@inmathwarn 112
\@input 101
\@input@ 101

\@item 295
\@itemdepth 299
\@itemitem 300
\@itemlabel 289
\@itempenalty 288
\@itemsep 283
\@iwhiledim 60
\@iwhilenum 60
\@iwhilesw 61
\@ixpt 175
\@kludgeins 471
\@labels 288
\@latex@error 69
\@latex@info 69
\@latex@info@no@line 69
\@latex@warning 69
\@latex@warning@no@line ... 69
\@latexbug 73
\@latexerr 70
\@lbibitem 404
\@listdepth 288
\@loadwithoptions 515
\@mainaux 93
\@makecol 435, 450
\@makefcolumn 434
\@makefnmark 590
\@makefntext 590
\@makespecialcolbox 451
\@maxdepth 442
\@minus 35
\@missingfileerror 102
\@mklab 295
\@multiplelabels 249
\@namedef 37
\@nameuse 38
\@needsformat 516
\@newcommand 40
\@newenv 45
\@newenva 44
\@newenvb 44
\@newl@bel 249
\@newline 81
\@newlist 289
\@next 440
\@nnil 61
\@no@lnbk 79
\@no@pgbk 79
\@nocnterr 71
\@nocounterr 71
\@nodocument 72
\@noitemerr 74
\@nolnerr 71
\@normalcr 80
\@notdefinable 71
\@notprerr 74
\@obsoletefile 102
\@onelevel@sanitize 54
\@onlypreamble 39
\@opcol 435, 449
\@options 514
\@outputpage 434, 453, 454
\@parmoderr 73
\@partaux 94
\@pkgextension 508
\@plus 36
\@popfilename 508
\@preamblecmds 39
\@preamerr 73
\@protected@testopt 42
\@providesfile 511
\@ptionlist 509
\@ptsize 567
\@pushfilename 508
\@qend 52
\@qrelax 52
\@rc@ifdefinable 43
\@reargdef 43
\@refundefined 248
\@reinserts 451
\@removeelement 62
\@reqcolroom 473
\@reset@ptions 519
\@resetactivechars 453
\@resethfps 475
\@roman 149
\@sanitize 54
\@savsf 82
\@savsk 82
\@scolelt 459
\@sdblcolelt 459
\@seccntformat 910
\@secondoftwo 48
@sect commands:
\@sect: 41, 42, 53
\@sect: 905
\@seqncr 271
\@setckpt 100
\@setfpsbit 475
\@setref 248
\@settodim 154
\@settopoint 154
\@slowromancap 149
\@spaces 70
\@specialoutput 447
\@specialpage 455
\@sptoken 53
\@sqrt 270
\@ssect 907

\@star@or@long 40
\@startcolumn 435, 458
\@startdblcolumn 435
\@stpelt 147
@tempa commands:
\@tempa: 185, 1652
\@tempboxa 154
\@testfp 474
\@testopt 41
\@textbottom 453
\@textfloatsheight 473
\@texttop 453
\@tfor 61
\@thirdofthree 48
\@toodeep 72
\@topnewpage 445
\@topsep 287
\@topsepadd 283, 287
\@totalleftmargin 288
\@trivlist 292
\@tryfcolumn 434, 459
\@twoclasseserror 521
\@twoloadclasserror 521
\@typeset@protect 51
\@unexpandable@noexpand ... 50
\@unexpandable@protect 50
\@unknownoptionerror 520
\@unusedoptionlist 508
\@use@ption 514
\@viiipt 175
\@viipt 174
\@vipt 174
\@vpt 174
\@vtryfc 459
\@warning 70
\@wckptelt 99
\@whiledim 60
\@whilenoop 60
\@whilenum 60
\@whilesw 60
\@whileswnoop 60
\@width 35
\@writeckpt 99
\@wtryfc 460
\@x@protect 51
\@xaddvskip 84
\@xargdef 40
\@xdblarg 54
\@xeqncr 271
\@xfootnotemark 384
\@xifnch 54
\@xiipt 175
\@xipt 175
\@xivpt 175
\@xnewline 80
\@xpt 175
\@xsect:n 908
\@xtryfc 460
\@xviipt 175
\@xxpt 175
\@xxvpt 175
\@yargd@f 42
\@yargdef 42
\@yeqncr 271
\@ytryfc 460
\@ztryfc 461
\[..... 268, 273
\{ 160
\addtolength 153
\newlength 153
\setlength 153
\settodepth 153
\settoheight 153
\settowidth 153
\} 160
\] 268, 273
\^ .. 170, 686, 688, 689, 2002, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2130, 2637, 2875, 2876, 2878, 2883, 2922
\= (¯) 780
\{ ({) 772
\} (}) 772
m 765
_(_) a772
_meaning 1751

\␣ 157, 236, 1183
\if@newlist 288
\if@noparitemfalse 289
\if@noparitemtrue 289
\set_borderline_aux:nn ... 894

A

\a 1
\AA 1
\aa 1
\AAA 1789, 1798
\aalph 211, 220, *685*
\abovedisplayskip 274
\abovedisplayskip *416*
\abstractname 11, 2354
\ABX 1929
accents 778, 779
acute (´) 778
arc (˘) 778

breve (˘) 778
caron (a˘) 778
\active@math@prime 267
\actualchar 1517, 1522, 2386, 2388, 2433, 2486, 2813, 2815
add commands:
\add_module_prefix:n 1796, 1801, 1804, 1807
add_to_callback 1
\add@accent 115
\add_to_callback *544*
\addcontentsline 397
\addcontentsline . 1, 72, 77, 1620, 1628, 2123, 2297, 2528, 2631, 2742
\addpenalty 85
\addpenalty 23
\AddPrefix 1802
\addto@hook 174
\addtocontents 1
\addtocounter 147
\addtolength 153
\addtolength 149, 150, 151
\addvspace 85
\addvspace . 24, *78*, *84*, 3027, 3142, 3150
\adp . . . 1935, 1942, 1945, 1946, 1947
\adsorbate 2161
\adsorbent 2162
\aegean 141
after commands:
\after_block:n 208
\afterassignment 2707
\aht 1935, 1941, 1944, 1947
\alldoztallies 2063
\allepsdice . . . 2018, 2019, 2020, 2021, 2022, 2023, 2033
\allocationnumber 1
\allowbreak 1, 914
\Alph 149
\alph 148
\alphabet 193
\alphabet *193*
\AlphAlph 361
\alphalph 354, 1697
\AlsoImplementation 1594
\alsoname 2370
\AltMacroFont 2557
\and *357*
\and 1
animals 768
\antimuon 2163
\antineutrino 2164
\antineutron 2165
\antiproton 2096, 2166
\antiquark 2167
\antiquarkb 2168
\antiquarkc 2169
\antiquarkd 2170
\antiquarks 2171
\antiquarkt 2172
\antiquarku 2173
\anyon 2174
\appendix 583
\appendixname 2357
\arabic 148
\arccos(arccos) a783
\arcsin(arcsin) a783
\arctan(arctan) a783
\Arg 447, *804*, *822*
\arg(arg) a783
arg commands:
\X_opt_arg 2440, 2447
arguments (environment) *806*
\arial 2834
Arial (font) 799
Arial Unicode MS (font) 204
\ARR 2214
\arrayrulecolor 1114, 1126
\arraystretch 1007
arrows 768
\ARRtrue 2216
asdf 727
\assumption 2175
\AtBeginDocument 520
\AtBeginDocument 139, 1145
\AtBeginDvi 442
\AtEndDocument 520
\AtEndDocument 2219, 2289, 2316, 2538
\AtEndOfClass 520
\AtEndOfPackage 520
\AtNextShipout 425
\AtNextShipout *425*
\atom 2176
atomic math objects 783
\attribute *541*
\aut@global 55
\author *357*
\author 1
auto 886
\auxm 3146, 3160
\awd 1935, 1940, 1943

B

background (package) 427
\backmatter 574
BAR 602

Basic Multilingual Plane 203
\begin 255
\begin 8, 17, 43, *255*, *274*, *299*, *328*, *335*, *499*, *589*, *728*, *730*, *731*, *732*, *741*, *744*, *746*, *754*, *755*, *756*, *789*, *797*, *798*
\belowdisplayskip . . . 3142, 3145
\bezier 1
\bfseries . . 48, 135, 194, 225, 316, 339, 393, 457, 488, 515, 859, 940, 983, 985, 986, 1207, 1224, 1248, 1483, 1531, 1707, 1737, 1915, 1955, 2473, 2493, 2509, 2535, 3021, 3024, 3042, 3046, 3050, 3058, 3154, 3175
\bgroup 1
\bibcite 404
\bibdata 405
\bibitem 404
\bibliography 405
bibliography font 205
\bibliographystyle 405
\bibliographystyle *403*
\bibname 13, 2355
\bibstyle 405
\Big . 2263, 2269, 2282, 2287, 2290, 2291, 2292, 2294
\big 263
\big 2453
\Bigassumption 2178
\bigassumption 2177
\bigbreak 1
\Bigg 2453
\biggassumption 2179
\Biggl 3216
\biggl 3216
\Biggm 3211
\biggm 3211
\Biggr 3222
\biggr 3222
\Bigl 3216
\bigl 3216
\Bigm 3211
\bigm 3211
\Bigr 3222
\bigr 3222
\bigskip 86
\bigskip 909, 1554, 1579, 1593, 2746
\bigskipamount 87
\blackacc . . 1767, 1937, 1951, 2331, 2349, 2426, 2430
\blackacchack . 1938, 1950, 2298, 2304, 2346, 2353, 2357, 2361, 2370
\blackacctwo 1952, 2437
\Bmesonminus 2180
\Bmesonnull 2181
\Bmesonplus 2182
\bmod 262
\bmod *262*
bobhook (package) 424, 425
\boldmath *166*
\bond 2183
book
class 566
environments
thebibliography 589
bool commands:
\bool_gset_false . . 24, 36, 43, 47, 51, 53, 57, 262, 264, 266, 268, 314, 315, 316, 323, 324, 325, 332, 333, 334, 340, 341, 342, 353, 354, 355, 362, 363, 364, 371, 372, 373, 379, 380, 381, 463, 465, 1595, 1980
\bool_gset_true . 23, 30, 31, 35, 39, 41, 45, 46, 49, 55, 89, 94, 98, 99, 103, 107, 141, 280, 281, 282, 283, 284, 459, 461, 1598
\bool_if 16, 21, 22, 110, 143, 166, 201, 204, 295, 310, 310, 318, 327, 336, 349, 357, 366, 375, 468, 472, 475, 479, 482, 487, 553, 571, 584, 679, 711, 722, 765, 771, 785, 787, 816, 965, 1020, 1023, 1039, 1040, 1054, 1059, 1071, 1076, 1109, 1128, 1292, 1294, 1300, 1330, 1353, 1431, 1432, 1474, 1602, 2092, 2106, 2128, 2236, 2363
\bool_lazy_all 1419
\bool_lazy_and . 1085, 2170
\bool_new 10, 11, 12, 13, 13, 14, 15, 16, 17, 18, 19, 22, 23, 26, 30, 31, 32, 33, 34, 35, 36, 37, 41, 42, 43, 44, 45, 53, 59, 71, 72, 73, 74, 75, 81, 98, 141, 261, 263, 265, 267, 1979
\bool_set 951
\bool_set_false 469, 547, 831, 836, 845, 924, 925, 926, 927, 928, 929, 930, 1168, 1180, 1185, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 2076, 2086, 2447
\bool_set_true . . 14, 76, 77, 476, 548, 825, 830, 837, 842, 843, 844, 850, 851, 860, 886,

887, 931, 1147, 1148, 1153, 1154, 1158, 1161, 1165, 1166, 1167, 1173, 1174, 1179, 1186, 1190, 1360, 2127, 2446
\bool_show 386
\c_false_bool 250, 769, 772, 1285, 2326
\c_true_bool 249, 1274
\boolcolor 1822
border-top-width 886
\bordermatrix 266
\boseDistrib 2184
\boson 2185
\botfigrule 483
\BOTLIST *435*
\botmark *420*
\bottomfraction 480
\bottomfraction *368*
\bottomrule 2374
\box 417
\box 2963
box commands:
\box_clear 1227
\box_gclear 921
\box_if_empty 918, 1104
\box_new 21, 65, 66
\box_use_drop ... 964, 1132
\box_wd 950
\box255 ..... *423*, *425*
\boxcolor 1835
\brace 264
\bracescolor 1911
\brack 263
\bracketscolor 1915
\break 22
\BreakableUnderscore 560
bricks (pattern) *791*
\bslash 2323, 2427
\buildrel 265

C

\c@bottomnumber 480
\c@dbltopnumber 481
c@enumi counter 299
c@enumii counter 299
c@enumiii counter 299
c@enumiv counter 299
\c@equation 269
c@equation counter 269
\c@errorcontextlines 70
\c@topnumber 480
\c@totalnumber 480
calc (package) 155
call commands:
call_callback 1
\call_callback *545*
callback commands:
callback_descriptions .. 1
\callback_descriptions ... *545*
\caption 371
\caption 2348, 2674
\captionsenglish 2376
\CaptionsList 2342
\carrow 2796
case 886
\cases 265
catcode commands:
\c_catcode_active_space_tl 180
\c_catcode_other_space_tl 749, 1341
\catcodetable *541*
\catcodetable@atletter ... *542*
\catcodetable@initex *542*
\catcodetable@latex *542*
\catcodetable@string *542*
\catcoding *111*
\cB 1756, 1768, 1781, 1874, 1883, 1904, 1923, 1934, 1966, 1976, 2009
\ccname 2366
\ccolor 1832
\cctabcolor 1841
\cE 1756, 1768, 1781, 1874, 1885, 1904, 1923, 1934, 1966, 1976, 2009
center (environment) 1
\centering 1, 324, 349, 2673
\centerline 1
\ch@ck 17
\changelogfont 3048
\changeloglabelfont 3050
\changes *843*
\chapter 912
\chapter 2, 3, 6, 347
chapter commands:
\chapter_open_any_bool 354, 363, 366, 381
\chapter_open_anywhere_bool 355, 364, 373, 375
\chapter_open_left_bool 349, 362, 371, 379
\chapter_open_right_bool 353, 357, 372, 380
\chaptername 885
\chaptername 17, 188, 2356
\chapterparent 890
\char .. 60, 154, 157, 160, 448, 1024, 1038, 1041, 1325, 1529, 1834,

1889, 2097, 2705, 2714, 2736, 2847, 2888, 3056
char commands:
\char_set_active_eq .. 934
\char_set_catcode 614
\char_set_catcode_active .. 170, 935, 2058, 2059, 2077, 2078
\char_set_catcode_letter 397, 398, 399
\char_set_catcode_other 2057, 2060, 2061
\char_set_lccode ... 2062, 2063, 2064, 2065, 2066
character 193
\charcolor 1819
\chardef@text@cmd 112
check marks 782
\check@command 47
\CheckCommand 47
\CheckEncodingSubset 137
\checkmark(✓) a782
\choose 263
\circle 1
\circledR(®) a782
\citation 404
\cite 404
\cite *403*
CJK 208
CJKV 208
\cl@@ckpt 148
\cl@page 245
class environment 789
\class 2728
\ClassError 67
\ClassError 1, 2284
classes
fontdef.dtx 262
kernel-ltmath.dtx 262
KOMA 567
KOMA 210
ltxdoc 390, 726, 746
memoir 210
Octavo 567
octavo 485
phddoc 390, 681, 721, 726
standard 566
\ClassInfo 67
\ClassWarning 67
\ClassWarningNoLine 67
\cleardoublepage 444
\cleardoublepage . 280, 320, 359
\clearpage 435, 444
\clearpage .. *94*, *98*, 173, 182, 203, *433*, *434*
\cleartoevenpage 311, 350
clist commands:
\clist_clear 932
\clist_count:N *810*
\clist_count:n *810*
\clist_gput_right 74
\clist_gset 79, 120, 130, 1729, 1761, 1889
\clist_map_function 1632, 1784, 1895, 2245, 2264
\clist_map_inline 114, 125, 179, 186, 415, 420, 492, 1061, 1625, 1732, 1738
\clist_new 10, 68, 75, 80, 111, 119, 129, 1786
\clist_put_right . 115, 1627
\clist_set 859, 2380
\g_docinput_clist . 10, *807*, *851*, 1627, 1632
\clistcolor 1817
\closeout 204, 1680
\cls 156, 452, *804*
\CLSLpig 2200
\CLSLpipe 1907, 1913, 1919
\clubpenalty *94*, 223, 229, 232, 236
\cmd .. 153, 420, 442, *803*, *812*, *825*, 1757, 2608, 2609, 2653, 3054
\cmdI . 1758, 1766, 1767, 2653, 2655, 2657
\cmdIp 1768, 2656
\cmdW 1767, 2656
\cmdX 1766, 2654
\cnt 3062
code2001 203
codecolorize 859
\CodedocExplain 512
\CodedocExplainEXP 512
\CodedocExplainREXP 512
\CodedocExplainTF 512
\codeexample 2908
\codeexamplebox 2907, 2929, 2940, 2963
\codeexamplenl 3000, 3011
\codeexamplewidth .. 2891, 2905, 2911, 2940, 2955, 2960
\codelinefont 1736
\CodelineIndex 145
CodelineNo 401
\CodelineNumbered 146
\codetwothousand 203
\codetwothousandone 203

coffin commands:
\coffin_clear 920, 922, 923
\coffin_join . 967, 971, 975, 984, 988, 992
\coffin_new ... 17, 18, 19, 20
\coffin_typeset .. 979, 996
\coffincolor 1834
coffinkeywords commands:
\coffinkeywords_clist 1761
\coffinkeywordsregex 1764
\col@number 443
\colOpt 1008, 3103, 3104
\color 14, 48, 103, 157, 208, 213, 214, 215, 222, 232, 239, 240, 241, 250, 262, 268, 269, 270, 279, 314, 315, 316, 322, 352, 360, 361, 362, 363, 364, 393, 401, 402, 403, 403, 404, 635, 651, 655, 663, 665, 680, 681, 709, 714, 718, 722, 726, 729, 736, 737, 738, 739, 803, 807, 812, 859, 884, 888, 893, 940, 978, 983, 984, 985, 986, 1023, 1031, 1039, 1156, 1534, 1703, 1707, 1738, 1746, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1867, 1915, 1955, 2360, 2558, 2708, 2936, 3024, 3154, 3175
\colorbox 2936, 2973
\colorizebraces 1916, 2016
\colorizebrackets .. 1907, 2018
\colorizecomments .. 2005, 2047
colorizecomments 1981
\colorizehandlers ... 1777, 2021
\colorizeifs 1752, 2046
\colorizemathoperator ... 1959, 2019
\colorizemodulekeywords . 1784, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2032, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045
\colorizesquarebrackets . 1927, 2017
colorizesquarebrackets .. 1907
\colorizexparsedeclarations 1969
\colorlet 2892, 2893
Colors
thecs a726
themeta a746
theoption a797
\columnsep . 1602, 1605, 2751, 2754
\columnseprule 1601, 2751
combo commands:
\combo_if_bool 141, 143
\command 2711
\comment 470, 477, *823*
\commenthandler 2884, 2900, 2933
\commentshack 2048
Computer Modern (font) 192
\conductivity 2186
\contentsline 1
\contentsname 591
\contentsname 2358
\continueLineNumber *731*
\continuelinenumber 731
\continuelinenumber 329
\copyright 214
\cos(cos) a783
\cosh(cosh) a783
\cot(cot) a783
\coth(coth) a783
Counters
c@enumi 299
c@enumii 299
c@enumiii 299
c@enumiv 299
c@equation 269
foocounter 747
incsyms 772, 779
totalsymbols 770
\covbond 2187
\create_callback *545*
\CS *37*
\cs 104, 153, 420, 443, 519, *803*, *803*, *803*, *807*, *812*, *825*, *842*, 2204, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 3054
cs commands:
\cs:w 17, 25, 28, 31, 34, 37, 39, 41, 44, 69, 70, 73, 76, 80, 91, 103, 112, 117, 118, 127, 146, 153, 156, 168, 170, 176, 187, 189, 261, 262, 273, 280, 281, 295, 296, 308, 309, 310, 315, 316, 317, 322, 323, 324, 327, 329, 330, 331, 336, 337, 338, 343, 344, 345, 352, 353, 354, 355,

360, 361, 362, 367, 368, 369, 374, 375, 376, 381, 382, 383, 388, 389, 390, 395, 396, 397
\cs_end 17, 25, 28, 31, 34, 37, 39, 41, 44, 70, 71, 73, 76, 82, 91, 103, 112, 119, 120, 129, 146, 153, 156, 168, 170, 178, 187, 189, 261, 262, 273, 280, 281, 295, 296, 308, 309, 310, 315, 316, 317, 322, 323, 324, 327, 329, 330, 331, 336, 337, 338, 343, 344, 345, 352, 353, 354, 355, 360, 361, 362, 367, 368, 369, 374, 375, 376, 381, 382, 383, 388, 389, 390, 395, 396, 397
\cs_generate_variant . 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 117, 123, 261, 606, 675, 775, 782, 1029, 1035, 1082, 1281, 1287, 1351, 1409, 1784, 1785, 1801, 2341, 2368
\cs_gset 2, 6, 216, 244, 259, 263, 276, 306, 309, 313, 320, 327, 334, 341, 348, 351, 358, 365, 372, 379, 386, 393, 395, 400, 405, 406, 1529, 1644, 1670, 1686, 1695, 1702, 1713, 1869, 1878, 2005, 2014, 2291
\cs_gset_eq ... 48, 407, 408, 1026, 1712, 1753, 1754, 1755, 1906
\cs_gset_protected . 2303, 2317
\cs_if_exist 115, 1000, 1333, 1715
\cs_if_exist:NTF . *804*, *805*
\cs_if_exist_p:N . *804*, *805*
\cs_if_exist_use 4, 5, 6, 7, 8, 68, 79, 160, 161, 162, 163, 164, 182, 183, 184, 185, 186, 215
\cs_if_free 1888
\cs_new ... 11, 15, 21, 78, 100, 110, 112, 162, 192, 198, 214, 216, 232, 234, 240, 242, 246, 253, 256, 272, 292, 296, 321, 327, 339, 346, 360, 361, 415, 435, 438, 800, 949, 984, 1083, 1205, 1796, 1844, 1865, 1876, 1907, 1932, 1933, 1969, 2542, 2546, 2556
\cs_new_eq 86, 87
\cs_new_nopar 70
\cs_new_protected 118, 134, 142, 164, 171, 262, 303, 308, 545, 607, 621, 625, 641, 644, 647, 660, 668, 676, 696, 703, 709, 727, 729, 739, 744, 763, 776, 801, 866, 876, 889, 901, 907, 911, 916, 937, 941, 945, 954, 959, 961, 1001, 1017, 1030, 1036, 1042, 1057, 1069, 1102, 1122, 1199, 1211, 1239, 1243, 1252, 1254, 1272, 1282, 1288, 1297, 1313, 1324, 1346, 1352, 1358, 1371, 1395, 1402, 1410, 1417, 1436, 1441, 1448, 1463, 1615, 2079, 2141, 2168, 2220, 2234, 2329, 2344, 2369, 2397, 2431, 2439, 2463, 2468, 2475, 2480, 2489
\cs_new_protected_nopar 940
\cs_set . 8, 8, 9, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 52, 53, 63, 89, 96, 108, 120, 131, 133, 138, 142, 144, 157, 172, 176, 177, 179, 206, 208, 222, 253, 256, 257, 258, 259, 260, 266, 283, 307, 314, 335, 378, 458, 505, 547, 591, 637, 680, 724, 767, 811, 853, 901, 917, 933, 1007, 1008, 1029, 1035, 1044, 1048, 1090, 1176, 1189, 1222, 1246, 1258, 1260, 1271, 1288, 1302, 1323, 1342, 1350, 1450, 1453, 1455, 1463, 1469, 1481, 1486, 1528, 1534, 1548, 1554, 1559, 1570, 1571, 1581, 1582, 1591, 1592, 1595, 1598, 1610, 1613, 1614, 1616, 1617, 1625, 1626, 1633, 1638, 1639, 1642, 1651, 1705, 1712, 1719, 1752, 1777, 1787, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1867, 1916, 1927, 1937, 1959, 1969, 1974, 1990, 1996, 2007, 2234, 2241, 2242, 2247, 2259, 2263, 2265, 2266, 2296, 2302, 2328, 2376, 2382, 2456, 2470
\cs_set_eq 18, 419, 442, 443, 444, 445, 1226, 1658, 1661, 1662, 1663, 1664, 1665, 1666,

1669, 1674, 1675, 1676, 1677, 1684, 1699, 1710, 1725
\cs_set_protected 244, 409, 2070
\csc(csc) a783
\cscolor 1813
\csname 648
\csname 1, *37*, 211, 230
\ctan 785
\ctan 2533
\curr@fontshape 168
currency symbols 765, 788
\currentfile . . 176, 195, 197, 198, 230, 231, 233, 1638, 1639, 1667, 1671, 1673, 1715, 1716, 1718
\CurrentOption 508
\CurrentOption 500
\CurrentOption 65, *518*
\currentsectionname 36, 146, 148
\cvaref 1007, 3101
\cxset 1, 5, 60, 63, 85, 114, 122, 142, 149, 150, 160, 162, 178, 186, 187, 195, 200, 214, 221, 238, 239, 262, 265, 265, 268, 271, 274, 278, 281, 292, 299, 300, 309, 334, 351, 359, 377, 394, 420, 437, 474, 507, 521, 564, 608, 654, 697, 723, 741, 783, 827, 869, 897, 1020, 1047, 1054, 1064, 1071, 1073, 1080, 1090, 1103, 1106, 1112, 1126, 1128, 1133, 1134, 1161, 1171, 1190, 1220, 1250, 1288, 1317, 1349, 1351, 1379, 1391, 1393, 1402, 1427, 1435, 1477, 1479, 1508, 1552, 1553, 1581, 1612, 1637, 1640, 1668, 1670, 1693, 1694, 1705, 1734, 1761, 1788, 1816, 1907, 1934, 1962, 1999, 2038, 2066, 2094, 2131, 2173, 2175, 2201, 2263, 2291, 2315, 2316, 2344
cypriote (script) 764

D

\dashbox 1
\date *357*
\date 1, *7*
\dateitalian 2373
\dateukrainian *1*
\day 144, 172, 217, 260, 325, 326, 373, 416, 456, 499, 545, 586, 631, 632, 676, 719, 763, 806, 849, 892, 1043, 1086, 1128, 1155, 1183, 1212, 1243, 1285, 1311, 1346, 1374, 1432, 1459, 1502, 1545, 1547, 1575, 1605, 1634, 1662, 1703, 1727, 1756, 1783, 1810, 1838, 1929, 1956, 1997, 2036, 2060, 2088, 2116, 2155, 2197, 2265, 2281, 2284, 2338
\dbend 1529, 1535
\dblfigrule 483
\dblfloatpagefraction 481
\dblfloatpagefraction *369*
\dblfloatsep 482
\dbltextfloatsep 482
\dbltopfraction 481
\dbltopfraction *368*
\dcircle 789, 798, 878
\dday 2260
\deadcycles *416*
\DEBUGOFF 37
\DEBUGON 17, 57
\declare . . 2707, 2724, 2736, 2847, 2853
\declare@robustcommand 50
\DeclareDocumentCommand . . . 2, 7, 121, 149, 179, 291, 331, 420, 423, 426, 432, 1382, 1392, 1393, 1519, 1563, 1573, 1583, 1594, 1597, 1600, 1612, 1623, 1631, 1633, 1727, 3226, 3230
\DeclareDocumentEnvironment 480, 491, 494
\DeclareEncoding *111*
\DeclareEncodingSubset 136
\DeclareEncodingSubset . . . *136*
\DeclareErrorFont 170
\DeclareExpandableDocumentCommand 416, 434, 436
\DeclareFixedFont 159
\DeclareFontEncoding 161
\DeclareFontEncodingDefaults 163
\DeclareFontFamily 160
\DeclareFontFamily 141, *160*
\DeclareFontShape 158
\DeclareFontShape 140, 142, 143, 144
\DeclareFontSubstitution . 162
\DeclareMathSizes 163
\DeclareMathSymbol . 2091, 2095, 2101
\DeclareOption 512
\DeclareOption 500

\DeclareOption 25, 28, 33, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 61, 63, 65
\DeclareOption* 512
\DeclareOption* 500
\DeclareOptionX 643
\DeclareOptionX* 643
\DeclarePreloadSizes 163
\DeclarePreloadSizes . . 18, 20, 21, 22, 25, 26, 27, 28, 29, 30, 34, 38, 44, 46
\DeclareProtectedCommand *91*, *91*
\DeclareRobustCommand 50
\DeclareRobustCommand 156, 452, 453, 455, 456, 1023, 1745, 2032, 2033, 2061, 2062, 2063, 2320, 2533, 2550, 2551, 2552, 3166
\DeclareSymbolFont 2090, 2093, 2100
\DeclareTextAccent 115
\DeclareTextCommand 112
\DeclareTextCommandDefault 114
\DeclareTextFontCommand . . 240
\DeclareTextSymbol 112
\def 153, 154, 167, 168, 176, 182, 211, 212, 219, 220, 242
\default@ds 512
\defaultnumberingsystem . . . 37
\defaultscriptratio 1
\defaultscriptscriptratio . . 1
\define@newfont 168
\definecolor 10, 11
\deflength 3071
\deg(deg) a783
\depthof 155
\DescribeOption 1486
\description 1545, 1562
\descriptionlabel 580
\det(det) a783
\detokenize . 60, 1024, 1717, 1899, 1900, 3091
\dim(dim) a783
dim commands:
\dim_compare_p 952
\dim_gzero_new 164, 165, 166, 167, 169, 170, 171, 172, 174, 175, 176, 177
\dim_if_exist 13
\dim_new 24, 1101
\dim_set 949, 1106
\dim_zero 947
\dimcolor 1818
\ding *202*, 645, 789, 798, 878, 1539, 2400, 2401, 2402, 2409, 2410
\dipole 2188
disable commands:
disable_callback 1
\disable_callback *545*
\DisableDocumentation 458
\DisableImplementation . . . 458
\displaylines 266
displaymath (environment) 1
\displaystyle . 2380, 2386, 3207
\displayverbfont 18
\Dmesonminus 2189
\Dmesonnull 2190
\Dmesonplus 2191
\do *9*, 185
\doc 1998
doc commands:
\g_doc_functions_seq . . 11, 1032, 2179, 2195
\g_doc_macros_seq 12, 1303, 2181, 2193
doc left 749
doc right 749
\docAuxCommand 726
\docAuxCommand 121
\docAuxEnv 757
\docAuxEnv 1473
\docAuxKey 758
\docAuxKey 1519
\docAuxListingsStyle 291
\docClass 784
\docClass 2508
\docColor 726
\docColor 149
\docColorBox 1315
docCommand environment 746
docCommands environment 798
\docCounter 747
\docCounter 1048
\docCounter* 747
docEnv* 756
\docExtension 1719
\docextension 1724
\docFile 763
\docFile 1712
\docFileExtension 2526
\docfileextension 785
\docfileextension . . 2520, 2526
\docFilename 2519
\docfilename 785
\docfilename 2514, 2519
\docFont 799
\docFont 3168
\docFontArial *799*
\DocInclude 167, 171, 1633

\docinclude *683*
\docincludeaux 174, 219, 1636, 1695
\DocInput 807
\DocInput 198, 1622, 1673
\DocInputAgain 1631
docKey environment 756
\docSymbol 787
\docSymbol 2660
\document 94
documentation (environment) . 466
\documentclass 514
\documentstyle 514
\docValue 727
\docValue 179, 3135
\docZ 783
\docZ 2456
\DoNotIndex *499*, 1292, 1299
\dospecials *259*
\dotfill 25
\doublecovbond 2192
\downarrow 2224
\DOZ 2053
\doz . 2061, 2062, 2064, 2065, 2066, 2067, 2068, 2069
\DOZe 2062
\DOZtrue 2055
\DOZx 2061
\dp 2558
draftwatermark (package) 425
\draw 22, 42, 2772, 2776, 2788, 2792, 2802, 2848, 2853
\DTMsetstyle *1*

E

\edef 184, *420*
\EDICE 2013, 2027
\EDICEtrue 2029
\egroup 1
\eject 1
\electron 2193
element commands:
\element_delimiter 110, 115
\else 180, 205, 218, 232
else commands:
\else: 181, 2254
\emph 315, 316, 1439
\emphasis 334
\emphasize 870
\empty 20
\empty 865
\EmptyPageContents ... 267, 271
\emptypagecontents 263, 268, 282
\EnableCrossrefs 147, 1638
\EnableDocumentation 458
\EnableImplementation 458
\encapchar .. 115, 1310, 1316, 1510, 1511, 1518, 1523, 1717, 1722, 1747, 1800, 1805, 1843, 1850, 1893, 1895, 1900, 1902, 2393, 2433, 2482, 2483, 2487, 2516, 2522, 2523, 2808, 2809, 2812, 2814, 2815, 2828, 2829, 2832, 2833, 3167
\encfive 2249, 2370, 2544
\encfour .. 2353, 2357, 2361, 2362, 2543
\encgreek 2250, 2251, 2545
\enclname 2365
encoding schemes 194
OML 194
OMS 194
OMX 194
OT1 194
U 194
encodings 128
\encone ... 2237, 2238, 2245, 2246, 2349, 2542
\end 1, 4, 23, 48, *274*, *299*, *328*, *335*, *416*, *522*, *589*, *728*, *730*, *731*, *732*, *741*, *744*, *746*, *754*, *755*, *756*, *789*, *797*, *798*
end commands:
\end: 708
\end@float 375
\endclass 2733
\endcodeexample 2999
\endcommand 2712
\endcomment 472, 479, *823*
\endcsname 1, 211, 230
\enddescription 1549, 1575
\enddispListing 1172
\enddocument 253
\enddocument *249*
\endenumerate 3246
\endfilecontents 521
\endfirsthead 2156
\endfoot 2159
\endgraf 20
\endgraf 505, 1534
\endgroup *285*
\endhead 2155
\endinput 169, *522*
\endlastfoot 2160
\endline 1
\endlist 293
\endlist .. 2556, 2733, 3082, 3092
\endlrbox 1
\endluaexample 963

\endmacroarguments 2546
\endmacroaruments 2556
\endmarglist 3099
\endminipage 313
\endofcodeexample 2926
\endpicture 1
\endsmallverbatim 3133
\endtabbing 1587
\endtcblisting 950, 953, 956, 959, 994, 997
\endtexexample 963
\endtextoken 2712
\endtheglossary . 194, 202, 1666, 1677
\endtheindex 1613, 2761
\endtrivlist 294
\endVerbatim 408
\endverbatim 259
\endverbatim 406
\enlargethispage 6
\enquote 536, 3169
\enskip 89
\enspace 78, 89
\ensuremath 272
\ensuremath . 622, 627, 638, 2564, 3181
enumerate environment 299
\enumerate 3243
\env 452, *804*, 3058
Environments
class 789
docCommand 746
docCommands 798
docEnv* 756
docKey 756
enumerate 299
eqnarray 274
equation 274
function 807
itemize 299
keymarglist 797
lrbox 307
macro 807
phd@doc@head 755
phdd_manual_entry 754
picture 333
picture 328, 335
quotation 278
scriptexample 728
sverbatim 730
texexample 741, 744
teXX 731
teXXX 732
thebibliography 589
verse 580
environments
picture 328
environments:
arguments *806*
center 1
displaymath 1
documentation 466
equation 1
function 491, *804*
implementation 466
macro 491, *805*
math 1
sloppypar 1
syntax 497, *804*
texnote 503, *805*
thebibliography *403*
variable 480, *805*, *806*
\epsdice .. 2018, 2019, 2020, 2021, 2022, 2023, 2031, 2032, 2034, 2035, 2036, 2037, 2038, 2039
eqnarray environment 274
equation environment 274
equation (environment) 1
\errorstopmode *539*, 2283
\errorsym 2194
escapable characters 772
\EStruename 1956, 1957
\etameson 2195
\etamesonprime 2196
euro signs 765, 788
\eval *248*
\evensidemargin 151
\everydisplay 1, *416*
\everymath 167
everypage (package) 425, 427
\everypar *285*, *294*, *416*
everyshi (package) 424, 425
\EveryShipout 425
\EveryShipout *425*
\example 789
\example 2715
\exciton 2197
\ExecuteOptions 500, 514
\ExecuteOptions 66
\ExecuteOptionsX 643
\exp(exp) a783
exp commands:
\exp_after 121, 127, 128, 248, 354, 361, 793, 1249, 1794, 2255, 2473
\exp_args 138, 260, 298, 411, 413, 592, 618, 692, 716, 728, 809, 879, 1065, 1079, 1293,

1309, 1481, 2408, 2486, 2487, 2491, 2492, 2520
\exp_last_unbraced .. 799, 1334, 2448
\exp_not .. 120, 233, 274, 276, 278, 282, 288, 290, 294, 299, 317, 589, 1075, 1329, 1452, 1794, 2244, 2348, 2471
\exp_not:n *816*
\expandafter 1, 153, 230
\expcolor 1826
\experimentalsym 2161
\ExplFileDate 1720
\ExplFileVersion 1721
\ExplMakeTitle *680*
\ExplSyntaxOff ... 7, 8, 15, 19, 19, 20, 50, 51, 59, 61, 81, 82, 87, 94, 121, 126, 126, 128, 139, 155, 169, 185, 196, 214, 242, 257, 261, 264, 274, 291, 305, 307, 312, 319, 344, 345, 378, 403, 433, 433, 453, 482, 504, 509, 513, 533, 1005, 1009, 1027, 1048, 1072, 1209, 1220, 1321, 1327, 1358, 1376, 1389, 1416, 1461, 1479, 1525, 1552, 1568, 1579, 1589, 1593, 1643, 1725, 1744, 1756, 1808, 1908, 1934, 1988, 2005, 2012, 2052, 2240, 2248, 2279, 2314, 2342, 2383, 2394, 2404, 2449, 2464, 2670, 3248
\ExplSyntaxOn . 1, 1, 1, 3, 4, 5, 7, 9, 9, 20, 21, 42, 51, 52, 59, 62, 62, 72, 84, 88, 95, 97, 122, 127, 129, 140, 156, 171, 197, 215, 243, 258, 262, 275, 285, 295, 306, 308, 313, 320, 345, 346, 404, 434, 434, 460, 483, 506, 510, 999, 1006, 1025, 1028, 1053, 1175, 1210, 1221, 1322, 1328, 1359, 1377, 1390, 1424, 1462, 1480, 1526, 1553, 1569, 1580, 1590, 1594, 1704, 1726, 1730, 1752, 1787, 1905, 1930, 1967, 1989, 2006, 2055, 2233, 2241, 2258, 2295, 2327, 2375, 2396, 2439, 2455, 2659, 3240
\externalsym 2162
\extrachap 6
\extractclass 2731, 2735
\extrafloats 15

F

\faBitcoin(฿) a788
\faBtc(฿) a788
\faCny(¥) a788
faDollar(faDollar) a788
\faEur(€) a788
faEur(faEur) a788
faEuro(faEuro) a788
\faGbp(£) a788
\faIls(₪) a788
faIls(faIls) a788
\faInr(₹) a788
faInr(faInr) a788
\faJpy(¥) a788
faJpy(faJpy) a788
\fakeusepackage 2044, 2059
\faKrw(₩) a788
\fancyvrbtrue 3
\faRmb(¥) a788
faRouble(faRouble) a788
\faRub(₽) a788
faRub(faRub) a788
faRuble(faRuble) a788
faRupee(faRupee) a788
faShekel(faShekel) a788
faSheqel(faSheqel) a788
\faTry(₺) a788
faTry(faTry) a788
faTurkishLira(faTurkishLira) a788
\faUsd($) a788
faUsd(faUsd) a788
\faViacoin(ⱥ) a788
\faWon(₩) a788
\faYen(¥) a788
\fbox 308
\fboxrule 308
\fboxrule 146
\fboxsep 308
\fboxsep 145
\FC 1927
\FCbar 2361, 2362
\FCfalse 2355
\feat 2501
\Ferli() a779
\Fermi() a779
\fermiDistrib 2163
\fermion 2164
\feyn 2082
\fi 180, 183, 185, 188, 204, 205, 209, 218, 238
fi commands:
\fi: 14, 193, 204, 236, 280, 297, 349, 392, 472, 519, 562, 606, 652, 695, 739, 781, 825, 867,

915, 931, 947, 1062, 1104, 2256, 2279
\figurename 2362
\filbreak 1
\file 452, *804*
file commands:
\g_file_curr_name_str 1400
\file_if_exist:nTF 75
\file_input:n 77
file extension
.def 106
file extensions
.ist a396, a748
.tex a785
.tfm a208
cls a485
\filecontents 521
\filedate 236
\filekey . 197, 221, 225, 1670, 1671, 1699, 1707
\filena 234
\filename 1719
files:
float.dtx a429
gind.ist a396
phddoc.cls a675
puenc.def a197
sample.ist a396
source2e.pdf a3
source2e a5
template_html.tpl a763
testfont.tex a207
\filesep 209, 210, 220, 1691, 1694, 1698
\fileversion 237
\fill 88
\fill 272, 276
\Finale 1594
\firstmark *420*
flippdf (package) 425
\FLIST *434*
\floatpagefraction 481
\floatpagefraction *368*
\floatsep 481
\floatsep *435*
\flushbottom 410
\fmtname 1, *515*
\fmtversion 1, *515*
\fnsymbol 149
\font . . . 1, 2, 3, 141, 632, 633, 1528, 2526, 3129
font name 199, 200
font name 200
fontdef.dtx (class) 262
\fontencoding 1, 2542, 2543, 2544, 2545
\fontfamily *197*, 1356, 2561
fontinst (package) 193
fonts
Arial a799
Arial Unicode MS 204
cmr 194
cmss 194
cmtt 194
code2000 203
code2001 203
Dunhill 206
family 194
Fibonacci 206
Karl Berry scheme 204
non-serif 191
pi fonts 203
serif 191
sizing commands 209
Symbola 203
wingdings 203
Zapf Dingbats 202
\fontseries 1356, 2561
fontspec (package) 208
FOO 602
\FOO *36, 37*
foo commands:
\foo_underscore:N . . . 1984
foocounter counter 747
\footins 1
footitle 756
\footnote 383
\footnote 224, *381*, *423*, 2536
\footnotechanges 224
\footnotemark 381, 384
\footnoterule 590
\footnoterule 1
footnotes
\footnoterule 590
kernel \footnoterule 382
minipage (kernel) 382
\footnotesep 1
\footnotesize 27, 47, 208, 213, 224, 239, 262, 268, *685*, 773, 1019, 1131, 1166, 1373, 1439, 1706, 1993, 1999, 2010, 2263, 2402, 2460, 2979
\footnotetext 381
\footnotetext 1, *382*
\foreword 3
\forewordname 3
format 886

format commands:
\format_block:nnnn 95, 206, 310
\format_display:nnnn . 93, 307
\format_hang:nn 63, 99, 102
\format_hang_inmargin:nn 96, 103, 104
\format_inline:nnn ... 314
\format_inmargin:nnn . 52, 106
\format_part_traditional:nn 107, 321
\format_traditional:nn 109, 346
\format_block:nnnn 897
\format_display:nn 901
\format_hang:nn 895
\format_hang_inmargin 896
\format_inmargin 895
\fox 193
\frac 270
\frame 1
\framebox 309
\frameit *423*
\frenchspacing 19
\frontmatter 573
function environment 807
function (environment) .. 491, *804*
function commands:
\function_one: *805*
\function_two: *805*
\fussy 412
\fvset 7

G

\g@addto@macro 519
\gcd(gcd) a783
\gdef 206, 224, 229
\GenericError 64
\GenericInfo 64
\GenericWarning 64
\GetFileInfo 233, 1718
\getlinechar 1
\getpagerefnumber ... 1526, 1535
\getridofats ... 2685, 2691, 2698
\getridtest 2686, 2687
\global 227, *440*
\glossary *400*
\glossary 1
\glossaryname 2371
\GlossaryParms 2526
\GlossaryPrologue 2523
\Gluon 2165
glyph 193
gobble 1080
\goodbreak 1
\graphene 2166
\graviton 2167
\graybox 770
\graybox ... 1935, 1950, 1951, 1952
group commands:
\group_begin . 102, 169, 189, 323, 334, 348, 368, 805, 1030, 1037, 2056, 2075
\group_end 105, 195, 195, 338, 341, 372, 375, 810, 1033, 1042, 2069, 2163
\groupcolor 1821

H

h, George Douros 204
hacks
escapechar 45
\hang *24*
\hangafter 1533, 3152
hanja 208
\HARM 2316
harmony (package) 779
\HARMtrue 2318
\hb@xt@ 36
\hbond 2168
hbox commands:
\hbox:n 1230
\hbox_gset 1112
\hbox_gset_end 1127
\hbox_set 1304
\hbox_set_end 1309
\hbox_unpack_clear . 1262, 1305
\hboxcolor 1836
hcoffin commands:
\hcoffin_set:Nn .. 948, 963
\headheight 1
\headsep 1
\headtoname 2367
\Hermaphrodite 24, 25
\hfill 239
\hglue 22
\hideoutput 1
\hideskip 1
\hidewidth 1
\Higgsboson 2169
\hline 1009, 1013
\hole 2170
\hom(hom) a783
\hookleftarrow 2225
\hookrightarrow 2226

\hphantom 264
\href 2473, 2493, 2509, 2535
\hrule 2961
\hrulefill 25
\hsize *423*, 2898
\hskip *378*
\hspace 78, 88
\hspace *78*
\hspace* 78
\hspace* *78*
\Huge 1
\huge 1
\hyperlink 642, 645, 649
\hyperref 759
hyperref (package) 196
\hyperref 1334, 1533, 1550
\hypersetup 1147
\hypertarget 516, 524, 532
\hyphenchar . 632, 633, 2526, 3129
hànzì 208

I

\ialign 1
\idotsint($\int\cdots\int$) a781
\idxboth 1749, 1753, 2015
\idxbothbegin 1750, 1754
\idxbothend 1751, 1755
\idxCTAN 2539
\idxlanguage 764
\idxlanguage 1745
\idxTBsyms 2468
\ier 324, 325
\if *499*
if commands:
\if_case:w 180, 223, 267, 284, 336, 379, 459, 506, 549, 593, 639, 682, 726, 768, 812, 854, 902, 918, 934, 1049, 1091, 2233, 2266
\if_dim:w 9, 175
\if@compatibility 507
\if@endpe 295
\if@eqnsw 270
\if@fcolmade 442
\if@filesw 94
\if@firstcolumn 442
\if@inlabel 288
\if@insert 442
\if@mparswitch 443
\if@multiplelabels 249
\if@newlist 289
\if@newlistfalse 289
\if@newlistfalse 289
\if@newlisttrue 289
\if@newlisttrue 289
\if@nmbrlist 289
\if@nobreak 81
\if@noitemarg 289
\if@noitemargfalse 289
\if@noitemargtrue 289
\if@noparitem 289
\if@noparlist 289
\if@noparlistfalse 289
\if@noparlisttrue 289
\if@partsw 94
\if@reversemarginpar 443
\if@specialpage 442
\if@twocolumn 443
\if@twoside 443, 568
\ifARR 2213
\IfBooleanTF 29, 124, 151, 181, 293, 1475, 1521, 1565, 1575, 1585
\ifcase 213
\ifdefined 1626
\ifdim 133, 2940
\ifDOZ 2052
\ifEDICE 2026
\iffancyvrb 3, 4
\ifFC 2355, 2356
IfFileExists 491
\IfFileExists 100
\IfFileExists . 3, 175, 1637, 2028, 2054, 2088, 2215, 2317
\ifHARM 2315
\ifhmode 2987, 3125
\IfNoValueTF . . . 1792, 1813, 1824, 1840, 1858, 1869, 1880, 1890, 1980, 2003, 2274, 2310, 2390, 2445
\ifnum 178
\ifodd 280
\IfRefUndefinedExpandable 1530
\ifshowsymtable 2109, 2110, 2139, 2624, 2625, 2646
\ifSVR 2085
\IfValueTF 41, 68, 1730
\ifVIET 2368, 2369
\ifvmode 56
\ifx 185, 230
\ignorespaces *283*
\ignorespacesafterend . . 1, 501
\iiiint($\iiiint$) a781
\iiint($\iiint$) a781
\iint($\iint$) a781
\immdiate *94*
\immediate 183, 187, 188, 204, 207
implementation (environment) 466

in commands:
in_callback 1
\in_callback *545*
\include 97
\include *93, 94*, *98*, 179, 1641
\IncludeInRelease 1
\includeonly 97
\includeonly *93, 94*
\incsyms 770
incsyms counter 772, 779
\incsyms ... 1932, 1957, 1964, 1965, 1971, 1976, 1992, 1998, 2009, 2074, 2082, 2200, 2236, 2244, 2249, 2250, 2251, 2254, 2255, 2257, 2261, 2268, 2281, 2286, 2290, 2291, 2292, 2294, 2298, 2304, 2330, 2344, 2345, 2349, 2351, 2353, 2357, 2360, 2370, 2378, 2384, 2399, 2409, 2413, 2414, 2416, 2421, 2422, 2423, 2424, 2426, 2430, 2434, 2435, 2437, 2440, 2441, 2450, 2451, 2453, 2458
\index 389
\index *400*
\index 1
index doc settings 749, 758
index format 763
\indexaccent 769
\indexaccent .. 1906, 2298, 2304, 2331, 2349, 2353, 2357, 2370, 2426, 2430, 2437
\indexboth 765
\indexboth 1753
\indexbothbegin 1754
\indexbothend 1755
\indexcommand 766
\indexcommand .. 1757, 1761, 1763, 1788, 1906, 1993, 1999, 2074, 2082, 2237, 2245, 2249, 2250, 2251, 2254, 2255, 2262, 2268, 2281, 2286, 2294, 2351, 2379, 2385, 2400, 2409, 2413, 2414, 2417, 2422, 2423, 2434, 2435, 2440, 2441, 2450, 2451, 2453, 2459, 2658, 2664, 2667
\indexcs 716
\indexcypriot 767
\indexcypriot .. 1810, 1964, 1965
\indexDing 781
\indexDing 2397
\indexentry 208, 1690
\indexenv 724
Indexing
customizing 396
doc 390
general 390
multi-page 390
Indexing 390
\indexlinearb 767
\indexlinearb 1837
\indexmany 764
\indexmany 1727
indexmany commands:
\indexmany: 1729, 1732, 1738
\indexmathcmd .. 1888, 3192, 3196, 3199, 3202, 3205, 3209, 3213, 3219, 3224, 3228, 3232, 3238
\indexname 2361
\indexoption 2719
\IndexPrologue 2293
\indexpunct 770
\indexpunct 1771, 1773, 1909, 2010, 2291, 2292, 2421, 2424, 2657
\indexsoutharabian 769
\indexsoutharabian 1876
\indexsoutharabian 1877
\indexstaves 767
\indexstaves 1821, 1971, 1976
\indexTextcomp 782
\indexTextcomp 2415
\indexugar 768
\indexugar 1855
\indexvarpersian 768
\indexvarpersian 1866
\inf(inf) a783
\input 101
\input *55*, *93*, *493*
\InputIfFileExists 101
\InstanceKey 1586
\InstanceSemantics 1587
int commands:
\int_case 2508
\int_compare ... 55, 111, 341, 343, 345, 349, 354, 778, 1047, 1233, 1337, 1414, 1454, 1473, 1476, 1640
\int_eval 1236
\int_gdecr 1308
\int_gincr 1306, 1932
\int_incr 1213, 1322
\int_new 25, 49, 50, 67, 85, 1931
\int_set ... 1471, 2282, 2285
\int_to_Alph_finnish . 949
\int_to_alph_finnish . 984
\int_to_Alph_slovenian 1844

\int_to_alph_slovenian 1876
\int_to_symbols .. 951, 986, 1846, 1878
\int_use ... 1006, 1691, 1933
\c_zero *871*
\intarraycolor 1842
\intcolor 1820
integrals 781
\integralsymbol 800
\integralsymbol 3203
\interaction 2171
\interdisplaylinepenalty . 263
\interfootlinepenalty 19
\interlinepenalty 86, 133, 138, 325, 350, 2984, 2987, 3122, 3125
\internalsym 2172
\intextsep 481
\ion 2173
\ionicbond 2174
iow commands:
\iow_char 686, 688, 689
\iow_close 2215
\iow_indent 2312, 2533
\iow_log 422
\iow_new 2164
\iow_newline 2173, 2184, 2189, 2198, 2204, 2226, 2227, 2228, 2243, 2249, 2254, 2256, 2267, 2272, 2273, 2275
\iow_now ... 2185, 2190, 2199
\iow_open 2174
\iow_term 2173, 2216
\iow_term:n *864*
\iowcolor 1824
\ishortstack 1
\item 295
\item *24*
\itemcalendaroption 2724
\itemindent 287
\itemindent 1334, 2730
\itemitem *24*
itemize 299
itemize environment 299
\itemsep 283, 287
\itemsep .. 1335, 1344, 2552, 3033, 3097
\iterate 1
\itshape . *197*, 652, 656, 1039, 3036

J

\jot 263
\Jpsimeson 2175

K

\K 772
\k *129*
kanji 208
\Kaonminus 2176
\Kaonnull 2177
\Kaonplus 2178
Karl Berry Scheme 204
\Kcyp 771
\Kcyp 1966
\KED 774
\KED . 2018, 2019, 2020, 2021, 2022, 2023, 2073
\keepsilent 494
\ker(ker) a783
kernel
lists
\item 282
\list 291
kernel-ltmath.dtx (class) 262
\kernel@ifnextchar 52
keymarglist environment 797
Keys
/errors/boolean expected .. 670
/errors/unknown choice value 670
/errors/unknown key 670
/errors/value forbidden ... 669
/errors/value required ... 669
/utils/exec 669
doc left 749
doc right 749
/foo/ footitle 756
/key/ font name 199
/phd/ bibliography font 205
/phd/ font name 199, 200
/phd/ font name 200
/phdd/ codecolorize 859
/phdd/ index doc settings 758
keys commands:
\keys_define . 538, 820, 1136
\keys_set 551, 894, 1203
\keyscolor 1833
\keyval 795
\keyval *795*, 3014
\keyvaluestyle 1743
\keywordsregex 1900
keywordsregex 1900

keywordsregex commands:
\keywordsregex_aux:n 1878, 1897
\Kfeyn 2082
\Kgr 2253
\kill 1591
\KN 2254
\KNbig 2255
\Knoidx 2257
KOMA (class) 210
\KOMAoptions 102
\Kp 2007
\Kpig 2200
\Kr( ) a779
Kroll 389
Kroll
Leon Kroll 389
\Ks 2234
\Kstav 771
\Kstav 1979
Kstav commands:
\Kstav_no_opt_arg 1974, 1982
\Kstav_opt_arg . . 1969, 1985
\Kt 2242
\Kv 777
\Kv 2249

L

l internal commands:
\l__phd_make_new_heading_keys:n . 198, 417, 423, 424, 425, 426, 427, 428, 429, 430
\l__phd_make_new_numbering_keys:n 296, 418
\l__phd_set_headings_key_defaults:n . 435, 494, 495, 496, 497, 498, 499, 500, 501, 502
\l@ngrel@x 40
\l_@@_make_new_heading_keys:n 885
\l_@@_make_new_numbering_keys:n 888
\l_@@_set_headings_key_defaults:n 891
\l_phd_months_abbreviated_ 10
\l_phd_months_narrow_ 10
\l_phd_months_wide_ 9
\label 249
\label 52, *247*, 865, 991, 1065, 1079, 1385, 1409, 1433, 1440, 2016, 2152, 2222, 2348
label commands:
\label_number_box 142, 239
\labelargcount 2542, 2547
\labelargcount 2541
\labelenumiv *298*
\labelitemi 298
\labelitemi 1
\labelitemii 298
\labelitemii 1
\labelitemiii 298
\labelitemiii 1
\labelitemiv 298
\labelitemiv 1
\labelsep 287
\labelsep 2544, 3079, 3089
\labelwidth 287
\labelwidth 3078, 3088
\language 292, 634
\LARGE 1, 274
\Large 1
\large 1, 457
\lastbox 187, 226
\LastDeclaredEncoding 162
\lastskip *84*
\LaTeX . 1, *90*, *92*, *104*, *145*, *152*, *157*, *176*
LaTeX kernel
spacing
\newline 80
LaTeX kernel classesFile n
ltlength.dtx 153
Latex kernel@normalsize 487
Latex kernel@setfontsize 487
\LaTeXe 1, *90*, *92*, *104*, *145*, *152*, *157*, *176*, 2355
\latexe 2575, 2582, 2590
\latexreleaseversion 1
\latrack 20
\ldots *5*, 352, 361, 370, 378
\leavevmode 24
\leftarrow 2227
\lefteqn 271
\leftline 315
\leftmargin 287
\leftmargin . . . 1333, 2549, 2730, 3080, 3090
\leftmargini 288
\leftmarginii 288
\leftmarginiii 288
\leftmarginiv 288
\leftmark 1
\leftrightarrow 2228
\leftskip *284*
\Lefttorque . . 244, 273, 367, 1017
\len 3060
\let 186, 189, 190, 191, 192, 193, 194, 196, 199, 200,

201, 202, 205, 210, 221, 226, 227, 241, *454*, *499*
\levelchar 1522, 2386, 2433, 2486, 2813, 2815
\lg(lg) a783
\lim(lim) a783
\liminf(lim inf) a783
limits 783
\limsup(lim sup) a783
\line 337
Linear B 768
Linear B
\BPamphora (𐄀) a768
\BParrow (𐄁) a768
\BPbarley (𐄂) a768
\BPbilly (𐄃) a768
\BPboar (𐄄) a768
\BPbronze (𐄅) a768
\BPbull (𐄆) a768
\BPcauldroni (𐄇) a768
\BPcauldronii (𐄈) a768
\BPchariot (𐄉) a768
\BPchassis (𐄊) a768
\BPcloth (𐄋) a768
\BPcow (𐄌) a768
\BPcup (𐄍) a768
\BPewe (𐄎) a768
\BPfoal (𐄏) a768
\BPgoat (𐄐) a768
\BPgoblet (𐄑) a768
\BPgold (𐄒) a768
\BPhorse (𐄓) a768
\BPman (𐄔) a768
\BPnanny (𐄕) a768
\BPolive (𐄖) a768
\BPox (𐄗) a768
\BPpig (𐄘) a768
\BPram (𐄙) a768
\BPsheep (𐄚) a768
\BPsow (𐄛) a768
\BPspear (𐄜) a768
\BPsword (𐄝) a768
\BPwheat (𐄞) a768
\BPwheel (𐄟) a768
\BPwine (𐄠) a768
\BPwineiih (𐄡) a768
\BPwineiiih (𐄢) a768
\BPwineivh (𐄣) a768
\BPwoman (𐄤) a768
\BPwool (𐄥) a768
\linebreak 79
\linethickness 335
\linethickness 1, *329*
\linewidth 287
\linewidth . 259, 288, 2627, 2898, 2944, 2948, 2949, 2955, 2969, 3008
\list 291
\list 2547, 2729, 3076, 3086
\listfigurename 591
\listfigurename 2359
\listfiles 103, 486
\listfiles *55*
\listi 596
listings commands:
\listings_code . 2014, 2094, 2108, 2130
Listings Styles
extended *731*
highlite *732*
simple *729*
simplex *730*
singleline *730*
\listparindent 287
\listtablename 2360
\llap 1
\ln(ln) a783
\LoadClass 514
\LoadClass 501
\LoadClass 88, 93, 97, 101, 106, 109, 115, 125
\LoadClassWithOptions 515
\LoadClassWithOptions *505*
\log(log) a783
log-like symbols 783
\longhookrightarrow 2229
\longleftarrow 2230
\longleftrightarrow 2224
\longmapsfrom 2225
\longmapsto 2226
\longrightarrow 2227
\loop 1
\lorem 11, 13, 15, 274
\loremlines 1
\Lpack 2497, 2513
\lphdparagraphnumberprefixtl 64
\lphdparagraphnumbersuffixtl 65
\lphdsubparagraphnumberprefixtl 66

\lphdsubparagraphnumbersuffixtl 67
\lphdsubsectionnumberprefixtl 60
\lphdsubsectionnumbersuffixtl 61
\lphdsubsubsectionnumberprefixtl 62
\lphdsubsubsectionnumbersuffixtl 63
\lq 1
lrbox 307
\lstdefinelanguage 190, 417, 471, 746
\lstdefinestyle .. 225, 256, 308, 344, 659, 1010, 1154
\lstloadlanguages 197
\lstnewenvironment .. 298, 303, 324, 379, 385, 731, 775
\lstset 202, 299, 304, 325, 329, 331, 335, 380, 381, 386, 387, 732, 776, 777
\ltindex 2147, 2148, 2149, 2150, 2151
ltxdoc (class) 390, 726, 746
\ltxsyntaxfont 3044
\ltxsyntaxlabelfont 3046
\ltxtoday 5
\luacmd 3135
\luaexample 962
\luafunction *541*
\luatex 2056

M

\m@ne 11
macro environment 807
macro (environment) 491, *805*
\macroarguments 2546
\MacroFont 1338
\MacroLongFont 1338, 1354
\MacroTopsep 1256
\magicequal 2549
\magicequalname 2550
\magicrbrack 2548
\magicVertname 2360, 2552
\magicvertname . 1914, 1920, 2551
\magnon 2179
\magstep 1
\magstephalf 1
Mahjong 204
main
xindy 398
\main 1703
\mainmatter 573
make commands:
\make_new_bool:n . 259, 271, 272, 273
\make_new_opening_keys:n 276, 288, 289, 290
\makeatletter 54
\makeatother 54
\makebox 304
\makebox 105, 106, 653, 2564, 3190, 3195, 3198, 3201, 3204, 3207, 3208, 3211, 3212, 3216, 3217, 3222, 3223, 3227, 3231, 3235, 3236
\makeglossary *400*
\makeglossary 1
\makeindex 389, 400
\makeindex *400*
\makeindex *400*
\makelabel 293, 298
\makelabel *298*
\MakePercentComment 1620
\MakePercentIgnore 1618
\MakePrivateLetters 723
\MakePrivateLetters 7, 395, 1491, 2290
\MakeShortVerb 139
\maketitle *357*
\MakeUppercase 150
\maltese(✠) a782
\manual 1528, 1529
\mapsfrom 2228
\mapsto 2229
\marg 747
\marg 159, 447, *804*, *822*, 1035, 2203
\marginpar 379
\marginpar *378*
\marginparpush *378*
\marginparsep . *378*, 974, 978, 995
\marginparwidth .. 149, *378*, *682*, 952, 978, 1110
\marglist 3096
\marglistfont 3036, 3081
\marglistsep .. 3064, 3071, 3079, 3089
\marglistwidth 3066, 3069, 3078, 3088
\margnotefont 3040
\margoptionfont 3038, 3091
\mark 1, *410*
\markboth 1
\markright 1
\marks 529
math (environment) 1

math commands:
\c_math_operator_regex 1950, 1963
\mathdollar 1
\mathellipsis 1
\mathhexbox 1
\mathindent 272
\mathoperatorcolor 1952
\mathpalette 264
\mathparagraph 230
\mathparagraph 1
\mathsection 230
\mathsection 1
\mathsterling 1
\mathstrut 265
\mathsurround *91*
\mathunderscore 1
\mathversion 166
\mathversion *166*
\matrix 265
\max(max) a783
\maxdimen 1
\maxwellDistrib 2180
\maya 2413
\mayadigit 2413
\mbox 304
\medbreak 1
\medmuskip 231
\medskip 86
\medskip .. 2633, 2674, 2916, 2995
\medskipamount 87
\medskipamount 320, 321, 372, 373, 411, 412, 742, 743, 978, 991, 1164, 1165
memoir (class) 210
\MessageBreak 64
\meta 746
\meta . 160, 162, 164, 432, 445, 448, 450, 451, *804*, *804*, 940, 1026, 1029, 1046, 1410, 3175
\metalbond 2181
\method 2182
\midrule 1088, 2351
\min(min) a783
\minimumgroupdigits 41
\minipage 1
mode commands:
\mode_if_math:TF . 565, 628, 938
\mode_leave_vertical: 1131
\Module 2557
\module_error *544*
\module_info *544*
\module_warning *544*
monetary symbols 765, 788
\MoonPha *765*
\month . 144, 172, 218, 262, 328, 374, 417, 443, 501, 544, 586, 633, 677, 720, 764, 807, 850, 893, 1044, 1087, 1128, 1155, 1183, 1213, 1244, 1272, 1311, 1341, 1375, 1418, 1460, 1502, 1531, 1575, 1605, 1634, 1663, 1690, 1727, 1756, 1783, 1810, 1838, 1929, 1956, 1984, 2023, 2060, 2088, 2116, 2156, 2198, 2266, 2284, 2339
\MoonPha 2414
msg commands:
\msg_error 723, 872, 882, 919, 1105
\msg_info 79, 2316, 2538
\msg_new 73, 362, 370, 373, 376, 381, 386, 389, 2309, 2530
\msg_warning 188, 1141
\msgcolor 1827
\multicolumn 1091, 1097, 2154, 2158
\multiput 330
\multiput 1, 1
\muon 2183
musical symbols 779
\myempty 2682, 2687, 2696
\mygobble 2683, 2701
\myprintocmmand 2714

N

\N 778
\name *36*
names commands:
\z_make_names 112, 127
\narrower 24
\nathdouble 2281, 2282
\nathtriple 2287
\Nbig 2294
\nbitem 298
\nearrow 2230
\NeedsFormat *515*
\Needspace 2112
\NeedsTeXFormat 515
\NeedsTeXFormat 1, 2, 5, *111*
\negthinspace 78, 89
\neutrino 2184
\neutron 2185
\new@command 40
\new@environment 44
\new_attribute *542*
\new_bytecode *542*
\new_chunkname *542*

\new_luafunction *542*
\new_whatsit *542*
\newattribute *541*
\newboolean 486
\newbox 1, 2907
\newcatcodetable *541*
\newcommand 40
\newcommand 157, 171
\newcount 12, 148
\newcounter 147
\newcounter 782, 785, 2541
\newdimen 1
\NewDocumentCommand 267, 458, 460, 462, 464, 512, 514, 522, 530, 1198, 1473, 1501, 1736, 1788, 1802, 1810, 1821, 1837, 1855, 1866, 1877, 1888, 1979, 2001, 2272, 2308, 2388, 2441, 2443, 2465, 2532
\NewDocumentEnvironment . 466, 473, 497, 503, 3241
\newenvironment 44
\newfontfamily 141
\newhelp 1
\newif 45
\newif ... 3, 7, 12, 1049, 1050, 1051, 2026, 2052, 2085, 2202, 2213, 2222, 2315, 2355, 2368, 2906
\newinsert 16
\newlabel 249
\newlanguage 1
\newlength 153
\newlength 1940, 1941, 1942, 3013, 3064, 3066, 3074
\newline 80
\newluabytecode *541*
\newluachunkname *541*
\newluafunction *541*
\newmarks 1
\newmuskip 1
\newpage 435, 444
\newpage *433, 434*
\newread 1, *37*
\newrobustcmd 128, 1211, 1216, 1227, 1232, 1236, 1241, 1251, 1833, 3058, 3060, 3062
\newsavebox 306
\newsavebox 2555
\newskip 153
\newskip 1
\newtcolorbox 198, 200, 1360
\newtheorem 1
\newwhatsit *541*
\newwrite 1
\nfss@catcodes 169
\nkonumber 1417, 1432
\Nn 778
\Nn 2280
\Nnt 2285
\nobreak 22
\nobreak *365, 384*
\nobreakdashes 87
\nobreakspace 87
\nobreakspace 330, 357, 1341
\nobreakspace␣ *88*
\nocite 405
\nocite *403*
\node 361, 362
\noexpand *382*
\nofiles 96
\noindent . 211, 352, 361, 370, 378, 909, 955, 957, 958, 996, 1438, 1532, 1621, 2626, 2715, 2716, 2943, 2959, 3189, 3194, 3197, 3200, 3203, 3206, 3210, 3215, 3221, 3226, 3230, 3234
\noindexing 1614, 2762
\nointerlineskip 1, 251
\nolinebreak 79
\nolinkurl 452
\nonfrenchspacing 19
\nonumber 271
\nopagebreak 79
\nopagebreak *76*
\noprotrusion 1
\normalbaselines 20
\normalbaselineskip 1
\normalcolor 1, *479*, 1911
\normalfont 214
\normalfont 316, 457, 488, 515, 1015, 1943, 1944, 1945, 2979, 3048, 3134
\normallineskip 1
\normallineskiplimit 1
\normalmarginpar 1, *378*
\normalsfcodes 96
\normalsfcodes *455*
\normalsize 243, 487
\normalsize 488, 515, 1015
\notpredefinedmessage ... 2580
\notpredefinedmessageABX 2588
\Np 2293
\nucleus 2186
\null 21
\number 154, 209
\numberline . 1, 73, 78, 2124, 2632
\numberlineat 731
\numberlineat 332, 333

\numexpr 789, 798, 878
\nwarrow 2224

O

\o@lign 25
\oalign 25
\oarg 747
\oarg 161, 447, *804*, 1044
\obeycr 89
\obeylines 21
\obeyspaces 21
\obeyspaces 1119
\occitanday 1545, 1549
\occitanmonth 1531, 1549
octavo (class) 485
\oddsidemargin 150, 3069
\of 270
\offinterlineskip 1
\Ohne(´) a779
\oldherm 24
\on@line 70
\onecolumn 444
\OnlyDescription 1594
\ooalign 25
\open *423*
openbib 601
\openout 187, 1659
\openup 266
\operator@font 230
\operator@font 262
\opt 2499
\option 797
\option 3104
\optionlistfont 3042
\OptionNotUsed 512
\optname 2504
\or 213, 214, 215, 216, 217, 218
or commands:
\or: 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 727, 728, 729, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1099, 1100, 1101, 1102, 1103, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278
\orbit 2187
\origDeclareMathSymbol . 2091, 2101
\origDeclareSymbolFont . 2090, 2100
\origepsdice 2031, 2032
\orignewcommand 2319, 2324
\origProvidesPackage ... 2045, 2046, 2048
\origSetSymbolFont . 2092, 2102
\othernumberingsystems 39
\outerparskip 287
\output 447
\output *423*
\oval 1
\overfullrule 412
\ovidius 359

P

\p@ 18
\PackageError 67
\PackageError 2861
\PackageInfo 67

Packages
anyfont 210
moresize 210
xkeyval 642
packages
background 427
bobhook 424, 425
calc 155
draftwatermark 425
everypage 425, 427
everyshi 424, 425
flippdf 425
fontinst 193
fontspec 208
harmony 779
hyperref 196
phd 191
phd-pkgmanager 427
picture 328
prelim2e 426
sectsty 910
textcomp 765
uncial 201
watermark 425
xkeyval 486
Packages:
amsthm *355*
babel *245*
biblatex *403*
bobhook *425*
datetime2 . . . . . *1*
Doc *498*
doc *390*, *499*, *675*, *721*, *724*, *843*, *850*, *851*
doc.sty *826*
DocStrip *493*
docstrip *390*, *493*
environ *724*
etoolbox *60*, *724*
everypage *424*
expl3 . *60*, *665*, *804*, *806*, *806*, *850*, *870*, *871*
fancyvrb *821*
fixltx2e *438*
fontspec *191*
fonttable *207*
hypdoc *784*, *867*
hyperref . *247*, *644*, *759*, *828*, *844*
imakeidx *397*, *398*
keyval *642*
kvoptions *644*
l3basics *817*
l3doc *805*, *806*, *807*, *853*
l3docstrip *803*, *803*
l3keys *870*
Lazarides *784*
listings *721*, *728*, *741*
ltxdoc *675*, *803*, *804*, *804*, *804*
makeidx *389*, *745*
makeindex *396*
minted *721*
multind *397*
natbib *403*
perpage *245*
pgf *155*, *675*, *724*
pgfkeys *642*
phd *6*, *675*, *721*
phd-colorpalette . *721*, *728*, *746*, *797*
phd-documentation *682*
phddoc *676*
pict2e *333*
picture *332*, *332*
pspicture *333*
refcount *724*, *759*
tcolorbox *303*, *724*, *724*, *724*, *728*, *741*, *886*, *897*
titlesec *489*
titletoc *489*
tufte-book *501*
verbatim *724*
xkeyval *642*, *642*
xkview *642*
xparse *821*, *841*, *904*
xpicture *333*
\PackageWarning 68
\PackageWarningNoLine 67
\pagebreak 79
\pagebreak *434*
\pagegoal *423*
\PageIndex 1639
\pagename 2368
\pagenumbering 245
\pagenumbering *245*
\pageref 247
\pageref *248*, 1481, 1541
\pagestyle 409
palette 886
\panunicode 17, 425, 427, 443, 1373, 1415, 1747, 2021, 2346
\paperheight 442
\paperwidth 442
\par *284*, *285*, *294*, *411*, *416*
\par@deathcycles 292
\paragraph 916
\paragraph 16, 485, 492

paragraph commands:
\paragraph_cntformat . 259
\paragraphmark 1
\parbox 24, 44, 224, *369*, 1704
\parfillskip *284*
\parg 163, 447, *804*
\parindent *916*, *916*
\parsep 287
\parsep . . . 1336, 2550, 3076, 3077, 3082, 3086, 3087, 3092
\parshape *277*, 2949
\parskip 274
\parskip *284*, *416*
\part 574, 911
\part 195, 285, *683*, 1667, 2295, 2525
\partname 167, 2364
\partopsep 283
\PassOptionsToClass 511
\PassOptionsToClass 501
\PassOptionsToClass *505*
\PassOptionsToClass 65, 67
\PassOptionsToPackage 511
\PassOptionsToPackage *505*
\PassOptionsToPackage 59
\path 122
\pattern . . 2819, 2821, 2822, 2823, 2837, 2839
\patternindex 2805
\patternindexinherentlycolored 2826
Patterns
bricks *791*
\pdfpagewidth 212
\pdfstringdefDisableCommands 418, 440
\pdfstringnewline 415
penalties
\interfootline 19
\penalty *434*
\person 2532
pgf 763
\pgfkeys 653
\pgfkeysalso 653
\pgfkeysalso 866, 992
\pgfkeysdeargs 655
\pgfkeysdef 655
\pgfkeysedef 655
\pgfkeysedefargs 655
\pgfkeysgetvalue 652
\pgfkeysifdefined 652
\pgfkeyslet 652
\pgfkeyssetvalue 648, 651
\pgfkeysvalueof 652
\pgfkeysvalueof 147, 149
\pgfmanualdeclare . . 2707, 2708
\pgfqkeys 653
\pgfqkeys 3001
\pgfqkeysalso 653
\phantom 264
\phantom 656
\phantomsection 1372, 1626, 2115, 2630
phd (package) 191
phd commands:
\phd_after_heading 91, 137, 178, 212, 216
\phd_brackets 1323, 1407, 1412
\phd_doc_do_description 1350, 1386, 1413, 1442
\phd_float_box 147, 196, 242, 245, 253
\phd_format_router . 85, 89
\phd_hang_from . . . 6, 65, 137
\phd_index_col . . . 141, 1302
\phd_index_env . 1271, 1408, 1466, 1636
\phd_index_key . 1431, 1486, 1505
\phd_index_key:n 1480
\phd_index_key_path 1288, 1436, 1508
\phd_index_print_c . . 130, 1189, 1200, 1218, 1232, 1243
\phd_index_val 157, 176
phd_index_val 156
\phd_manual_top 1342
\l_phd_months_abbreviated_finnish 901
\l_phd_months_narrow_finnish 917
\l_phd_months_wide_abbreviated 283
\l_phd_months_wide_albanian 378, 417
\l_phd_months_wide_basque 505, 544
\l_phd_months_wide_brazil 547
\l_phd_months_wide_breton 591, 633
\l_phd_months_wide_bulgarian 637, 677
\l_phd_months_wide_catalan 680, 720
\l_phd_months_wide_croatian 724, 764
\l_phd_months_wide_czech 767, 807

\l_phd_months_wide_danish 811, 850
\l_phd_months_wide_estonian 853, 893
\l_phd_months_wide_finnish 933, 1044
\l_phd_months_wide_french 266, 328
\l_phd_months_wide_friulian 1048, 1087
\l_phd_months_wide_galician 1090
\l_phd_months_wide_german 222, 262
\l_phd_months_wide_greek 179, 218
\l_phd_months_wide_italian 335, 374
\l_phd_months_wide_russian 458, 501
\phd_more_keywords_tl 1787, 1797, 1798
\phd_print_col 133, 140, 146
\phd_print_com . 1205, 1382
\phd_print_env . 1222, 1407, 1412, 1465, 1471
\phd_print_key . 1430, 1435, 1481, 1504, 1507, 1514, 1516
\phd_print_key:n 1480
\phd_ref_doc 1528, 1556, 1570, 1581
\phd_ref_doc_star .. 1548, 1561, 1571, 1582
\phd_set_opening . 309, 348
\phd_setparent 405, 408, 409, 410, 411, 412, 413, 414

phd internal commands:

__phd_book_divisions_clist .. 114, 179, 415, 420, 492
\l__phd_chapter_afterskip_tl 394
\l__phd_chapter_align_tl 303, 402
\l__phd_chapter_beforeskip_tl 392
\l__phd_chapter_color_tl 403
\l__phd_chapter_fontfamily_tl 300, 399
\l__phd_chapter_fontshape_tl 302, 401
\l__phd_chapter_fontsize_tl 301, 400
\l__phd_chapter_fontweight_tl 299, 398
\l__phd_chapter_indent_tl 390
\l__phd_chapter_label_fontfamily_tl 166
\l__phd_chapter_label_fontshape_tl 167
__phd_create_new_element:nn 162, 181, 188
__phd_document_type_tl 69, 72
__phd_document_types_clist 68, 74
\l__phd_paragraph_afterskip_tl 497
\l__phd_paragraph_align_tl 504
\l__phd_paragraph_beforeskip_tl 496
\l__phd_paragraph_color_tl 505
\l__phd_paragraph_fontfamily_tl 501
\l__phd_paragraph_fontshape_tl 503
\l__phd_paragraph_fontsize_tl 502
\l__phd_paragraph_fontweight_tl 500
\l__phd_paragraph_indent_tl 495
\l__phd_paragraph_number_after_tl 253, 259
\l__phd_part_afterskip_tl 295
\l__phd_part_beforeskip_tl 293
\l__phd_part_color_tl 304
\l__phd_part_indent_tl 291
\g__phd_saved_heading_tl 83, 194
__phd_secondary_clist 129, 130, 186
__phd_secondary_clist_names_clist 119, 120, 125
\l__phd_section_afterskip_tl 419
\l__phd_section_align_tl 427
\l__phd_section_beforeskip_tl 417
\l__phd_section_color_tl 428

\l__phd_section_fontfamily_tl 424
\l__phd_section_fontshape_tl 426
\l__phd_section_fontsize_tl 425
\l__phd_section_fontweight_tl 423
\l__phd_section_indent_tl 415
__phd_set_document_divisions .. 78, 82, 88, 94, 100, 106
__phd_set_document_type:n 70, 77
\l__phd_subparagraph_afterskip_tl 523
\l__phd_subparagraph_align_tl 527
\l__phd_subparagraph_beforeskip_tl 522
\l__phd_subparagraph_color_tl 528
\l__phd_subparagraph_indent_tl 521
\l__phd_subsection_afterskip_tl 444
\l__phd_subsection_align_tl 448
\l__phd_subsection_beforeskip_tl 443
\l__phd_subsection_color_tl 449
\l__phd_subsection_indent_tl 442
\l__phd_subsubsection_afterskip_tl 469
\l__phd_subsubsection_align_tl 477
\l__phd_subsubsection_beforeskip_tl 468
\l__phd_subsubsection_color_tl 478
\l__phd_subsubsection_fontfamily_tl 473
\l__phd_subsubsection_fontshape_tl 475
\l__phd_subsubsection_fontsize_tl 474
\l__phd_subsubsection_fontweight_tl 472
\l__phd_subsubsection_indent_tl 467
phd-pkgmanager (package) 427
phd@doc@head environment ... 755
\phd_after_heading: 909
\phd_book_divisions_clist 882
\phd_create_new_element:nn 884
\phd_document_type_tl 881
\phd_float_box:nnn 900
\phd_format_router:nnn ... 906
\phd_index_col 753
\phd_index_env 753
\phd_index_key 753
\phd_print_com 751
\phd_ref_doc 759
\phd_secondary_clist 882
phdcl internal commands:
\g__phdcl_article_bool 12, 94
\g__phdcl_book_bool 11, 89
\g__phdcl_checkfunc_bool 41, 43
\g__phdcl_checktest_bool 45, 47
\g__phdcl_cs_break_bool 57
\g__phdcl_doctype_tl . 60, 62, 64, 85, 125
\g__phdcl_kernel_bool 49, 51
\g__phdcl_ldoc_bool ... 18
\g__phdcl_ldociii_bool 19
\g__phdcl_lmodern_bool 53, 55
\g__phdcl_macrocode_colorize_bool 10, 22
\g__phdcl_masterthesis_bool 17
\g__phdcl_report_bool 13, 98
\g__phdcl_scrartcl_bool 15, 107
\g__phdcl_scrbook_bool 14, 103
\g__phdcl_scrreprt_bool 16
\phdcolor 1838
\phdcs 747
\phdcs 1023, 1200, 1207, 1384, 1406, 1411, 1592
phdd commands:
\phdd_code_colorize_bool . 23, 24, 39, 1979, 2076, 2086, 2092, 2106, 2127, 2128
\phdd_doc_color_aux:n 127
\phdd_doc_color_star:n 127
\phdd_index_key:n ... 1480
\phdd_index_typeset_bool 97, 110

\phdd_key_color_color_tl 68, 130, 135, 1253
\phdd_key_color_command_tl . 64, 103, 1007, 1200, 1207
\phdd_key_color_environment_tl ... 65, 1213, 1218, 1224
\phdd_key_color_tl 66, 1229, 1232, 1483
\phdd_print_col 127
\phdd_typeset_bool 99
\phdd_typeset_command 100, 118, 1531
\g_phdd_typeset_documentation_bool 30, 35
\g_phdd_typeset_implementation_bool 31, 36
\phdd_typeset_index_command 108, 123

phdd internal commands:

__phdd_base_form_aux:nnN 268, 292, 318, 717
__phdd_base_form_aux:nnnnnN 306, 308
__phdd_base_form_signature_do:nnn 303
\g__phdd_base_name_tl 78, 747, 748, 754, 760, *812*
\g__phdd_checkfunc_bool 42, *809*, 2171
\g__phdd_checktest_bool 43, *809*, 1421, 2236
__phdd_cmd:nn 421, 424, 428, 545, *821*
\l__phdd_cmd_index_tl 69, 540, 549, 586, 589, *812*
\l__phdd_cmd_module_tl 70, 541, 550, 593, 596, *812*
\l__phdd_cmd_noindex_bool . 71, 542, 547, 584, *812*, *825*
\l__phdd_cmd_replace_bool ... 72, 543, 548, 553, *812*
\l__phdd_cmd_tl 68, 552, 556, 557, 558, 568, 570, 577, 580, 581, 588, 592, *812*
\g__phdd_cs_break_bool 45, 46, 571, *809*
\l__phdd_date_added_tl 82, *813*, 853, 1086, 1089, 1092
__phdd_date_compare:nNnTF 325, 879
__phdd_date_compare_aux:nnnNnnn 325
__phdd_date_compare_aux:w 325
__phdd_date_compare_p:nNn 325
__phdd_date_set:Nn . 853, 854, 866, 878
\l__phdd_date_updated_tl 83, *813*, 854, 1087, 1095, 1098
__phdd_deprecated_on:n 855, 876, 1194
\l__phdd_descr_coffin 19, *807*, 920, 956, 977, 980, 990
\g__phdd_design_colors_tl 1810, 1811, 1846
__phdd_detect_internals:N .. 164, 690, 2088, 2102, 2124, 2144, 2154
__phdd_detect_internals_aux:N 164
\l__phdd_detect_internals_bool 13, 14, 166, *807*
\l__phdd_detect_internals_cs_tl 16, 184, 189
\l__phdd_detect_internals_tl . 15, 173, 174, 175, 176, 177, 179, 181, 182, 184, 185, 186, 187, 190, *807*
\l__phdd_doc_def_tl . 2165, 2176, 2183, 2207
\l__phdd_doc_undef_tl 2166, 2177, 2188, 2210
__phdd_ensuremath_sb:n 607, *826*
\g__phdd_finale_tl . 1605, 1613, 1614
\g__phdd_func_iow .. 2164, 2174, 2185, 2190, 2199, 2215
__phdd_function:nnw 484, 492, *835*, 889
__phdd_function_assemble: 904, 961
__phdd_function_descr_start:w 899, 954
__phdd_function_descr_stop: 903, 954
__phdd_function_end: 489, 493, *835*, 889
__phdd_function_extra_labels: 1010, 1057
__phdd_function_index:n 1017
__phdd_function_init: 892, 916
__phdd_function_label:nN 1021, 1069, 1082

\l__phdd_function_label_clist .. 80, *812*, 859, 932, 1061
__phdd_function_reset: 898, 941
__phdd_function_typeset: 897, 945
__phdd_function_typeset_start: 891, 907
__phdd_function_typeset_stop: 905, 907
\l__phdd_functions_coffin .. 18, *807*, *808*, *834*, *835*, *836*, 923, 948, 950, 973, 994
__phdd_functions_typeout:nN .. 2202, 2208, 2211, 2220
__phdd_get_function_name:n ... 236, 240, 1045, 1278
__phdd_get_function_signature:n 238, 240
__phdd_get_hyper_target:nN . 668, 1063, 1073, 1327, 1478
__phdd_gprop_name:n . 360
__phdd_if_almost_str:nTF .. 109, 117, 264, 568, 2400
__phdd_if_detect_internals_ok:N 196
__phdd_if_detect_internals_ok:NTF 164
__phdd_if_macro_internal:nTF 783, 1302, 1348
__phdd_if_macro_internal_aux:w 783
__phdd_if_macro_internal_p:n 783, 1427, 1467
\l__phdd_in_function_bool 22, *808*, 931, 1128
\l__phdd_in_implementation_bool 73, 469, 476, 482, 487, *812*, *823*
\l__phdd_index_escaped_key_tl .. 2343, 2371, 2372, 2387
\l__phdd_index_escaped_macro_tl 2342, 2373, 2391
\l__phdd_index_internal_bool 59, 603, *811*, 2339, 2446, 2447
\l__phdd_index_key_tl 57, 599, *811*, *869*, *870*, 1310, 2335, 2442, 2445, 2450, 2452, 2455, 2458, 2465, 2466, 2477, 2482, 2485, 2492, 2494, 2505, 2519
\l__phdd_index_macro_tl . 56, 600, *811*, *869*, 2336, 2441, 2443
\l__phdd_index_module_tl .. 58, 595, 601, *811*, *869*, *870*, 2333, 2337, 2444, 2459, 2472, 2473, 2485, 2496, 2497, 2502, 2506, 2519
__phdd_input:n 1615, 1628, 1632
\c__phdd_iow_mid_rule_tl 63
\c__phdd_iow_midrule_tl *811*, 2226, 2228
\c__phdd_iow_rule_tl 61, *811*
\c__phdd_iow_separator_tl 2173, 2216
\g__phdd_kernel_bool . 44, 204, *809*
__phdd_key_drop_underscores: 2468, 2484, 2501
__phdd_key_func: *869*, *870*, 2453, 2480
__phdd_key_get:n 592, 2331, 2439
__phdd_key_get_base:nN 262, *869*, 2441
__phdd_key_get_base_TF:nN 266, 272
__phdd_key_get_module: 2489
__phdd_key_pop: 2451, 2463, 2478, 2483, 2499, 2500
__phdd_key_trim_module:n *814*, 2468, 2486, 2487, 2520
__phdd_key_var: . *869*, *871*, 2456, 2489
\g__phdd_lmodern_bool 41, *809*
\l__phdd_long_name_bool . 23, *808*, *835*, 951, 965, 1109
__phdd_lseq_name:n . 360, 728, *810*, *829*
__phdd_macro:nnw 483, 495, 1199
\l__phdd_macro_argument_tl .. 84, *813*, 883, 893, 1202
\l__phdd_macro_box 65, *811*, *844*, 1228, 1265, 1315, 1317
\l__phdd_macro_documented_tl 40, *809*, 1197, 1225, 1245, 1249
__phdd_macro_dump: 1208, 1254
__phdd_macro_end: .. 488, 496, 1410
__phdd_macro_end_check_tested: 1413, 1417
__phdd_macro_end_style:n 1415, 1436

__phdd_macro_end_wrap_item:n 1441
\l__phdd_macro_EXP_bool .. 35, *809*, 830, 836, 844, 927, 1039, 1167, 1179, 1185, 1219
__phdd_macro_index:nN 1293, 1297
\l__phdd_macro_index_box 66, *811*, 1227, 1262, 1304, 1305
__phdd_macro_init: . 1201, 1211
\l__phdd_macro_int 67, *811*, 1233, 1236, 1322
\l__phdd_macro_internal_bool .. 31, 787, *809*, *831*, 886, 1147, 1153, 1214
\l__phdd_macro_internal_set_bool . 30, 785, *809*, 887, 1148, 1154, 1215
\l__phdd_macro_noTF_bool .. 34, 771, *809*, 850, 926, 1174, 1218
\l__phdd_macro_pTF_bool . . 33, 765, *809*, 842, 925, 1166, 1217, 1431
__phdd_macro_reset: 1209, 1239
\l__phdd_macro_rEXP_bool .. 36, *809*, 831, 837, 845, 928, 1040, 1168, 1180, 1186, 1220
__phdd_macro_save_names: 1206, 1243
__phdd_macro_save_names_aux:n 1246, 1252
__phdd_macro_single:nNN . *827*, *841*, 1274, 1285, 1288
\l__phdd_macro_tested_bool 26, *808*, 1190, 1222, 1360, 1423
\l__phdd_macro_TF_bool 32, 722, 773, *809*, 816, 825, 843, 851, 924, 1161, 1165, 1173, 1216, 1432
__phdd_macro_typeset_block:nN 780, 1272
__phdd_macro_typeset_one:nN 1291, 1313
__phdd_macro_typeset_variant_list:gN 1277, 1282, 1287
\l__phdd_macro_var_bool . 37, *809*, 1158, 1221, 1422, 1474
__phdd_macroname_prefix:n 1342, 1346, 1351
__phdd_macroname_suffix:N 1343, 1352
__phdd_meta:n ... 433, 607
__phdd_meta_original:n 607
\g__phdd_missing_tests_prop 27, *808*, *846*, 1398, 2239, 2258
\g__phdd_module_name_tl . 60, 136, 139, 191, 198, 201, *811*, 1617, 2148
__phdd_names_block_base_map:N 801, 1246
\l__phdd_names_block_tl 51, 698, 731, 733, 741, 803, *810*, *829*, *830*, *831*, *835*, *842*
__phdd_names_get_seq:nN 676, 895, 1204
__phdd_names_parse: 696, 896, 1205
__phdd_names_parse_aux:Nnn 728, 729
__phdd_names_parse_aux:nnn 714, 716, 724, 727
__phdd_names_parse_one:n 696
__phdd_names_parse_one_aux:nnNn 706, 709
\l__phdd_names_seq 54, 700, *810*, 895, 1204, 1399, 1426
__phdd_names_typeset: 739, 1011, 1207
__phdd_names_typeset_auxi:n 739, *830*
__phdd_names_typeset_auxii:n 754, 759, 763, *830*
__phdd_names_typeset_block:nN 767, 772, 773, 776
\l__phdd_names_verb_bool . 53, 679, *810*, 863, 930, 1195, 1223
\l__phdd_nested_macro_int .. 25, 778, *808*, *831*, *841*, 1213, 1414
\g__phdd_nested_names_seq 55, *810*, 1248, 1253, 1451, 1465, 1466, 1469, 1472, 1479, 1484
\l__phdd_no_label_bool 81, *812*, 860, 929, 1059, 1071
\g__phdd_not_tested_seq . . 28, *808*, 1428, 2261, 2277
\l__phdd_output_coffin 17, *807*, 968, 972, 976, 979, 985, 989, 993, 996
\l__phdd_override_module_tl ... 38, 39, *809*, 864, 933, 943, 1196, 1224, 1241, 2332, 2333

__phdd_pdfstring_cmd:w 434
__phdd_pdfstring_cs:w 434
__phdd_pdfstring_meta:w 434
__phdd_pdfstring_newline:w 415
__phdd_predicate_from_base:n 234, 768
__phdd_print_documented: 1441
__phdd_print_end_definition: 1415, 1441
__phdd_print_macroname:nN 1320, 1324
__phdd_print_testfile:n 1226, 1358
__phdd_print_testfile_aux:n 1358
__phdd_quote_special_char:N 2372, 2429, 2431
\l__phdd_regexes_clist 1786
__phdd_replace_at_at:N . 134, 557, 691, *825*, 2089, 2103, 2125, 2145, 2156
__phdd_replace_at_at_aux:Nn 134
__phdd_shorthand_meta: 934, 937
__phdd_shorthand_meta:w 937
__phdd_show_functions_defined: 2168, 2219
__phdd_show_not_tested: 2234, 2289
__phdd_signature_base_form:n 214, 300
__phdd_signature_base_form_aux:n 214
__phdd_signature_base_form_aux:w 214
__phdd_special_index:nn .. 1033, 1307, 2306, 2329
__phdd_special_index_aux:nnnnnn 2342
__phdd_special_index_module:nnnnN ... 598, 2321, 2334, 2342
__phdd_special_index_set:Nn 2342
__phdd_split_function_auxi:w 240
__phdd_split_function_auxii:w 240
__phdd_split_function_do:nn 240, 268, 305, 705
\l__phdd_store_tl 1733, 1757, 1769, 1782, 1875, 1886, 1905, 1924, 1935, 1945, 1967, 1977, 2010, 2049, 2087, 2088, 2089, 2095, 2101, 2102, 2103, 2109, 2112, 2123, 2124, 2125, 2131
__phdd_str_if_begin:nnTF 124, 198, 201, 206
__phdd_syntax:w . 498, 1101
\g__phdd_syntax_box .. 21, *808*, *836*, 918, 921, 964, 1104, 1112, 1132
\l__phdd_syntax_coffin 20, *807*, *808*, 922, 963, 969, 986, 997
\l__phdd_syntax_dim . 1101, 1106, 1117
__phdd_syntax_end: . 500, 1101
__phdd_test_missing:n 1394, 1395
__phdd_test_missing_aux:Nnn 1397, 1402, 1409
\g__phdd_testfiles_seq 29, *808*, 1363, 1365
__phdd_tmp:w 120, 121
__phdd_tmpa:w 86, 244, 260, 2241, 2258, 2263, 2277, 2470, 2473
\l__phdd_tmpa_int . 49, *810*, 1471, 1473, 1476, 2282, 2285
\l__phdd_tmpa_seq . 50, 175, 176, 177, 806, 807, 808, *810*, 1426, 1430, 1450, 1454, 1456, 1459, 2504, 2508, 2512, 2514
\l__phdd_tmpa_tl .. 47, 266, 268, 609, 615, 618, 678, 682, 685, 687, 688, 689, 690, 691, 693, 746, 747, 750, 751, 807, *810*, 813, *826*, *828*, *831*, 878, 880, 884, 1064, 1065, 1078, 1079, 1332, 1333, 1334, 1339, 1340, 1342, 1404, 1405, 1406, 1407, 1443, 1444, 1446, 1480, 1481, 2143, 2144, 2145, 2149, 2158, 2159, 2238, 2252, 2270, 2280, 2284
__phdd_tmpb:w ... 86, 2245, 2247, 2264, 2265
\l__phdd_tmpb_tl .. 48, 555, 556, 558, 613, 615, 758, 760,

808, *810*, 814, 815, *831*, 2152, 2154, 2156, 2158
\l__phdd_trial_width_dim 24, *808*, 947, 949, 952, 1110
__phdd_trim_right:Nn 118, 179, 181, 185, 186, 187
__phdd_typeset_aux:n 640, 1045, 1349
__phdd_typeset_dates: 1012, 1083
\g__phdd_typeset_documentation_bool 74, 76, 459, 463, 468, 472, *812*
__phdd_typeset_exp: . 517, 640, 1039
__phdd_typeset_expandability: 1024, 1036, 1055
__phdd_typeset_function_block:nN 779, 1017
__phdd_typeset_functions: *835*, 948, 1001
\g__phdd_typeset_implementation_bool 75, 77, 461, 465, 475, 479, *812*, 1595, 1598, 1602, 2171
__phdd_typeset_rexp: 525, 640, 1040
__phdd_typeset_TF: . 533, 640, 1023, 1054, 1353
__phdd_typeset_variant_list:nN 1026, 1042
\l__phdd_undoc_def_tl 2167, 2178, 2197, 2213
\l__phdd_variants_prop 79, *812*
\g__phdd_variants_seq 52, 753, 757, 758, *810*, *830*, 1025, 1047, 1048, 1051, 1275, 1284
__phdd_xmacro_code:n *853*, 1727, 2071, 2079
__phdd_xmacro_code:w 1727, 2082, 2121, 2141
phdd_manual_entry environment 754
\phddcolor 1839
phddoc (class) 390, 681, 721, 726
phddoc commands:
\phddoc_orig_meta ... 1025
\phdidxfont 3166, 3170
\phdindexcom ... 1260, 1383, 1635
\phdindexprintca 1176, 1213, 1229, 1238, 1253
\phdindexprintcolc ... 127, 1315
\phdindexprintcolca . 1251, 1309
\phdindexprintcomc . 1198, 1267, 2516
\phdIndexPrintCs 1799, 1833, 1850
\phdindexprintenvc .. 1216, 1284
\phdindexprintenvca . 1211, 1277
\phdindexprintkeyc . 1232, 1295, 1298, 1496
\phdindexprintkeyca 1227, 1291, 1489
\phdIndexPrintValC ... 170, 1241
\phdindexprintvalca .. 163, 1236
\phdlabelprefix 43, 52
\phdmakedocSubKey 1450
\phdwritetemp 1171
\phimeson 2188
\phimesonnull 2189
\phonon 2190
\photon 2096
\pickup@font 167
picture 333
picture environment 328, 335
\pigpenfont 2200
\pionminus 2191
\pionnull 2192
\pionplus 2193
\pkg 452, *804*, 1772, 2359, 2492
\pkgname 783
\pkgname .. 1927, 1928, 1929, 2027, 2053, 2086, 2214, 2316, 2472, 2497, 2513, 2583, 2584, 2591, 2592, 2593
\plasmon 2194
\plotmarkentry 2844
\plotmarkentrytikz 2850
\pmatrix 265
\pmod 263
\polariton 2195
\polaron 2196
\positron 2197
\Pr(Pr) a783
\preface 2
\prefacename 2, 15, 2352
prelim2e (package) 426
prg commands:
\prg_break 276, 278, 282, 288
\prg_break_point 290
\prg_generate_conditional_variant 132
\prg_new_conditional 325, 783
\prg_new_protected_conditional 109, 124, 196
\prg_return_false 114, 130, 208, 210, 346, 350, 355, 788, 796

\prg_return_true . 115, 129, 199, 202, 207, 346, 350, 355, 788, 796
\prgcolor 1823
\prime@s 267
\printanat 2690, 2705
\PrintChanges 191, 192, 200, 1663, 1664, 1675, 2537
\PrintDescribeEnv 1499
\PrintIndex . . 189, 190, 199, 1661, 1662, 1674, 2315
\printindex 389, 745
\printproperties 192
\ProcessKeysPackageOptions 124
\ProcessOptions 512
\ProcessOptions 123, *518*
\ProcessOptions* 500, 512
\ProcessOptionsX 644
prop commands:
\prop_get 106, 1404
\prop_get:NnNTF 88
\prop_gput 108
\prop_gput:Nnn 88
\prop_if_empty 2239
\prop_map_function . . 2257
\prop_new 27, 79
\prop_put 107, 1407
\prop_put:Nnn 88
\propcolor 1825
\protect 52
\protect *382*
\protected@edef 51
\protected@write 97
\protected@xdef 51
\protein 2161
\proton 2162
\provide@command 47
\providecommand 46
\providecommand 159, 161, 163, 449, 450, 451, 1354
\ProvideFile *107*
\provides_module *544*
\ProvidesClass 511
\ProvidesFile 511
\ProvidesFile 486
\ProvidesFile 3
\ProvidesPackage 510
\ProvidesPackage 499
\ProvidesPackage . 2, 6, *111*, 2045, 2046, 2048
\ProvideTextCommand 112
\ProvideTextCommandDefault 114
\ps@empty 409
\ps@plain 409
\PSfont 2527
\put *342*

Q

\Q 778
\qbezier 1
\qbeziermax 1
\Qc 779
\Qc 2328
\qcolor 1831
\Qe 780
\Qe 2343
\Qiv 2357, 2365
\QivBAR 2360, 2366
\Qpc 2351, 2365, 2366, 2373
\Qpfc 2353
\qquad 89
\qquad 2017
\Qt 2349
\quad 89
\quad 89, 231, 250, 251, 252, 253, 254
\quadrupole 2163
Quantum Mechanics
History 389–390
\quark 2164
quark commands:
\q_mark 249, 250, 254, 255, 257, 326, 328
\q_no_value 39, 549, 550, 933, 943, 1224, 1241
\quark_if_no_value 586, 593, 2332
\q_stop 120, 121, 215, 232, 251, 254, 255, 257, 326, 328, 358, 2082, 2121, 2141, 2470, 2473
\quarkb 2165
\quarkc 2166
\quarkcolor 1828
\quarkd 2167
\quarks 2168
\quarkt 2169
\quarku 2170
quotation environment 278
\quotechar 2433, 2436
\Qv 2370, 2373

R

\R 781
\r@@t 264
\raggedbottom 410
\raggedbottom *410*
\raggedleft 1
\RaggedRight 293

\raggedright 1, 225, 457, 1118, 1707, 2960, 3098
\raisebox 516, 524, 532, 2351, 2558
\rbrack 20
\ref 247, 248
\ref *248*, 1534, 1550
ref commands:
\ref_com: 1553, 1566
\ref_com_star . . . 1559, 1566
\refAux 761
\refAux 1591
\refAuxcs 1592
\refCmd 760
\refCmd 1563
\refEnv 1573
refEnv commands:
\refEnv: 1570, 1577
\refEnv_star 1571, 1576
\reference 2171
\refKey 760
\refKey 1583
\refname 2353
\refstepcounter 249
\refstepcounter 60, *247*, 865, 991, 2116, 2119, 2629
regex commands:
\c_bool_regex 1855
\c_box_regex 1860
\c_braces_regex 1907, 1920
\c_brackets_regex 1909, 1941
\c_char_regex 1856
\c_clist_regex 1850
\c_comments_regex . . 2000, 2007
\c_cs_regex 1790, 1794
\c_dim_regex 1853
\c_exp_regex 1852
\c_group_regex 1859
\c_handlers_regex 1776, 1779
\c_ifs_regex 1750, 1754
\c_int_regex 1857
\c_iow_regex 1858
\regex_const . . . 1750, 1776, 1790, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1907, 1908, 1909, 1950, 2000
\c_regex_regex 1863
\regex_replace_all . . 574, 1753, 1765, 1778, 1785, 1871, 1880, 1901, 1918, 1929, 1939, 1961, 1970, 2007
\regex_replace_once . . 869
\c_seq_regex 1849
\c_skip_regex 1861
\c_str_regex 1854
\c_tex_regex 1851
\c_tl_regex 1848
\c_token_regex 1862
regex internal commands:
__regex_compile:n . . 1791
\l__regex_internal_regex 1794
\regexcolor 1814
\registernumber *543*
registers
1,3,5,7,9 10
allocation 10
\relax 1, 2, 3, 154, 172, 188, 190, 192, 196, *214*, 227, 230
remove commands:
remove_from_callback . . . 1
\remove_from_callback *544*
\removeats 2695
\removelastskip 1
\renew@command 43
\renew@environment 45
\renewcommand 43
\renewcommand 20, 286, 347, 411, 438, 454, 463, 485, 492, 512, 517, 1007, 2093, 2094, 2095, 3081, 3091
\renewenvironment 44
\repeat 1
\RequirePackage 514
\RequirePackage 486
\RequirePackage 2, 3, 5, 27, 130, 131, 132, 134, 135, 136, 137, 1003, 1392, 1727, 2057
\RequirePackages *493*
\RequirePackageWithOptions 515
\resistivity 2172
\restore@protect 51
\restorecr 89
\restoresymbol 2323
\returntospace 2878, 2932
\reversemarginpar 1, *378*
\rhomesonminus 2173
\rhomesonnull 2174
\rhomesonplus 2175
\rightarrow 2225
\rightline 1
\rightmargin 287
\rightmargin 1337
\rightmark 1
\rightskip *284*
\Righttorque . 243, 272, 366, 1016

\rlap 1
\rmfamily 1031, 1039
Robertson, Will 207
\Roman 148
\roman 148
\romannumeral 1212, 1218
\root 264, 270
\rootbox 264
\rq 1
\rule 1935

S

\samepage 80
\samepage 3032
\sanitize 766
\sanitize . 1790, 1812, 1823, 1839, 1857, 1868, 1879
\sanitized 1790, 1796, 1799, 1805, 1812, 1815, 1818, 1823, 1826, 1829, 1839, 1850, 1857, 1860, 1863, 1868, 1871, 1874, 1879, 1882, 1885
\sarrow 2780
\savebox 306
\savebox 2557
\savesymbol 2321
\sb 266
\sb 622
\sbox 307
\sbox *24*
scan commands:
\scan_stop: 127, 128, 189, 714, 724, 751, *810*, 814, *829*, 1528
\scantokens 1834, 2860, 2937
scriptexample environment .. 728
scripts:
cypriote 764
\scriptsize 1
\searrow 2226
\sec(sec) a783
\secdef 1
\section 914
section commands:
\section_cntformat ... 256
\section_number_after_tl 247, 250, 256
\section_number_after_tl . 910
\sectionmark 1
\sectionname 885
sectsty (package) 910
\seename 2369
\selectfont *161*
seq commands:
\seq_clear 681
\seq_clear_new 734
\seq_count . 1047, 1454, 1472, 2508
\seq_gclear 753, 1484
\seq_get 750, 808
\seq_gpop 758
\seq_gput_right .. 89, 1032, 1248, 1253, 1303, 1365, 1428
\seq_gset_eq 757
\seq_gset_filter 1465
\seq_if_empty .. 1025, 1275, 1469, 2261
\seq_if_in . 1363, 2181, 2195
\seq_item .. 1459, 1479, 2512, 2514
\seq_map_function 699, 2276
\seq_map_function:NN . *841*
\seq_map_inline 1284, 2179, 2193
\seq_map_variable 177
\seq_new 11, 12, 28, 29, 52, 54, 55
\seq_pop 746, 807
\seq_pop_left 176
\seq_put_right 682, 735, 737
\seq_set_eq 806
\seq_set_filter 1426
\seq_set_from_clist .. 692
\seq_set_map 1450
\seq_set_split 88, 175, 2504
\seq_set_split:Nnn ... 88
\seq_use ... 1048, 1051, 1399, 1430, 1456
\seq_use:Nnnn *847*
\seqcolor 1815
set commands:
\set_borderline_aux:nn 21, 210, 254
\set_borders_aux:nn 11, 26, 32, 38, 42, 218, 219, 220, 221
\set_color:nn 335, 369
\set_color_aux:nn . 15, 216, 217
\set_font_aux:n 2, 67, 78, 102, 107, 126, 148, 272, 294, 326, 328, 336, 351, 355, 362, 370, 422, 447, 471, 499, 525
\set@display@protect 51
\set@typeset@protect 51
\set_color_aux:nn 894
\set_font_aux:n 894
\setattribute *542*
\setboolean 486

\setcaptions 4, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121
\setcounter 146
\setcounter 152
\sethyperref 133
\setkeys 2912
\setlength 153
\setlength 148
\setrefcountdefault 1526
\SetSymbolFont 2092, 2094, 2102
\setto 154
\settodepth 1945
\settoheight 154
\settoheight 1944
\settowidth 154
\sffamily .. 26, 46, 405, 3154, 3175
\sh@ft 25
\shipout 421, 455
\shipout *98, 422, 423, 425*
\shortstack 335
\showaccent 801
\showaccent 3230
\showhyphens 174
\showlbrace 800
\showlbrace 3215
\showmbrace 800
\showmbrace 3210
\showop 800
\showop 3206
\showoutput 1
\showover 3234
\showoverfull 1
\showrbrace 801
\showrbrace 3221
\showrelsymbol 800
\showrelsymbol 3200
\showrover 801
\showsymbol 800
\showsymbol 3194
\showsymbolalpha 799
\showsymbolalpha 3187
\showsymbolbin 800
\showsymbolbin 3197
\showwideaccent 3226
\sigma 1701
\sin(sin) a783
\sinh(sinh) a783
\skewchar 141
skip commands:
    \skip_gset:Nn 11
    \skip_vertical:N .. 89, 136
    \l_tmpa_skip 7, 9, 11, 24, 174, 175, 177, 197, 198, 211
\slash 22
\SLiTeX *91*
\sloppy 410
sloppypar (environment) 1
\smallbreak 1
\smallskip 86
\smallskip . 1679, 2715, 2716, 3163
\smallskipamount 87
\smallverbatim 3132
\smallverbatimsize 3134
\smash 265
\solid 2176
\sourceformat 660
\sp 266
\space 20
\space 179
spacing
    \hglue 22
    \vglue 22
    french 19
    non-french 19
``special'' characters .. 772
\SpecialIndex 2303
\SpecialMainEnvIndex ... 1636
\SpecialMainIndex .. 50, 1635
\SpecialOptionIndex ... 1501, 1505
\SpecialPackageIndex 784
\SpecialPackageIndex ... 295, 2478, 2494
\spin 2177
\spindown 2178
\spinup 2179
\split@name 168
\splittopskip 419
\sqrt 270
\sqrtsign 270
\sqrtsign 270
\Square() a772
square commands:
    \c_square_brackets_regex 1908, 1931
\squarebracketscolor ... 1913
\ss 674
\stackrel 269
\star 642
start commands:
    \start_section:nnnnnnnnn ... 2, 48, 289, 387

\start_section:nnnnnnnnn 904
\startlineat 731
\startlineat 331, 332
\startnumberat 333
\stepcounter 147
\stewart 121
\stok 721, 723
\stok 3180
\stop 1
\StopEventually 1594
str commands:
\c_backslash_str ... 412, 424, 430, 589, 672, 813, 2413, 2426, 2450
\c_percent_str 686
\str_case 90, 218, 2377, 2494
\str_case:nn 88
\str_case_e 85
\str_case_x 91
\str_count 1337
\str_head 2494
\str_if_eq 298, 751, 814, 2352, 2374
\str_tail 2492
\c_underscore_str ... 671
\strcolor 1829
\stretch 89
\string ... 14, 94, 111, 153, 179, 183, 208
\stripcommand 2697, 2701
\strippedat 2688, 2690, 2696
\strippedtext 2697, 2698
\strut 1, 224
\strutbox 1
\subitem 590
\subparagraph 916
\subparagraph 512, 517
subparagraph commands:
\subparagraph_cntformat 260
\subparagraph_number_after_tl 254, 260
\subparagraphmark 1
\subsection 914
\subsection ... 12, 438, 454, 1542, 1558, 2208
subsection commands:
\subsection_cntformat 257
\subsection_number_after_tl 251, 257
\subsectionmark 1
\subsubitem 590
\subsubsection 915
\subsubsection 14, 463, 1550, 1559, 1576, 1582, 1583, 1589
subsubsection commands:
\subsubsection_cntformat 258
\subsubsection_number_after_tl 252, 258
\subsubsectionmark 1
\sup(sup) a783
\supereject 23
\suppressfloats 473
\surface 2180
sverbatim environment ... 730
\SVR 2086, 2146
\svrexample 2181
\svrphoton 2182
\SVRtrue 2089
\swarrow 2227
\symarrow 2766
\symbola 203
symbols
currency 765, 788
log-like 783
monetary 765, 788
musical 779
variable-sized 781
\symbolsdecimal 43
\symbolsexponential 55
\symbolsgroup 45
\symbolsinfinity 61
\symbolslist 47
\symbolsminussign 53
\symbolsnan 59, 63
\symbolspermille 49
\symbolsplussign 51
\symbolssuperscriptingexponent 57
\symbolstimeseparator 65
syntax (environment) ... 497, 804
sys commands:
\c_sys_jobname_str 2174, 2313, 2534, 2535
\syscolor 1843

T

\tabbing 1590
\tabcolsep 1008
\tablename 2124, 2363, 2619, 2632
\tableofcontents 587
\tachyon 2183

\tan(tan) a783
\tanh(tanh) a783
\task 168, 242
\tauleptonminus 2184
\tauleptonplus 2185
\TC 2576
\tcbdimto 1365, 1366
\tcbdocmarginnote 54
\tcbIndexPrintComC 114
\tcblisting .. 950, 952, 955, 958, 994, 996
\tcblower 1354
\tcbox .. 109, 148, 154, 174, 191, 198
\tcbset 23, 29, 38, 44, 55, 186, 788, 791, 871, 964, 1159, 1160, 1644
\tcbuselibrary 28
\Tding 2408
\TemplateArgument 1546
\TemplateKey 1563, 1566
\TemplateSemantics 1547, 1573
\tencirc 3
\tencircw 3
\tenln 2
\tenlnw 2
\testfile 1226, 1392
\TestFiles 806
\TestFiles 806, 1382
\TestMissing 806
\TestMissing 806, 1393
\TestSections 18
\testsections 926
\testsections 8, 18
\TeX ... 1, 90, 92, 104, 145, 152, 157, 176
\tex 1080
TEX and LATEX 2ε commands:
\@ 515, 825
\@@@@par 3118, 3125
\@@endpbox 1
\@@input 55
\@@line 1
\@@par 284
\@@startpbox 1
\@DEBUGfalse 13, 38
\@DEBUGtrue 18
\@M 1, 86, 133, 138, 188, 223, 325, 350, 1544, 1561, 1585
\@MM 1, 57
\@Mi 1
\@Mii 1
\@Miii 1
\@Roman 316, 330
\@aalph 211, 212
\@addtoreset .. 165, 166, 402, 783, 786
\@afterheading 1
\@afterindentfalse .. 12, 17
\@afterindenttrue . 1, 8, 18
\@arabic . 103, 337, 344, 368, 375, 382, 389, 396, 790
\@author 1
\@auxout .. 93, 178, 186, 205, 247, 1640, 1658, 1684
\@begin@tempboxa 1
\@begintheorem 1
\@bezier 1
\@bitor 437
\@botlist 435
\@car 248
\@carcube 1
\@cclv 1
\@cclvi 1
\@cdr 248
\@centercr 1
\@checkend 1
\@circ 1
\@circle 1
\@clnht 1
\@clnwd 1
\@clubpenalty .. 200, 232
\@combinefloats 435
\@compatibility 517
\@ctrerr 218
\@currdir 7
\@currentlabel . 51, 247, 1384, 1409, 1432, 1438
\@currenvir ... 522, 2374
\@currenvline 1
\@dashbox 1
\@dashcnt 1
\@dashdim 1
\@date 1
\@dbflt 1
\@dblfloat 1
\@dblfloatplacement .. 1, 432
\@declaredoptions ... 518
\@dischyph 564

\@docinclude .. 180, 182, 1642, 1644
\@documentclasshook . 517
\@dot 1
\@dottedtocline 1
\@downline 1
\@downvector 1
\@eha 179, 1641
\@empty 210, 256
\@end@tempboxa 1
\@endpbox 1
\@endtheorem 1
\@evenfoot 226, 241, 1710, 1725
\@filelist 1
\@filesw 93, 94
\@finalstrut 1
\@floatboxreset 1
\@floatpenalty 1
\@flushglue 1
\@footnotemark 1
\@footnotetext 382
\@for 184, 1652
\@framebox 1
\@freelist 435
\@getcirc 1
\@getlarrow 1
\@getrarrow 1
\@gobble 221
\@gobble@till@return 2884, 2885, 2899
\@gobble@till@return@ignore 2886, 2899
\@halfwidth 1
\@hangfrom 1, 6
\@highpenalty 1
\@hline 1
\@hvector 1
\@icentercr 1
\@iden 1
\@idxitem ... 1610, 1611, 2759, 2760
\@ifnextchar 1966, 2253, 2289, 2293, 2433, 2452
\@ifpackagelater 506
\@iframebox 1
\@ignorefalse 1
\@ignoretrue 1
\@iiiminipage 1
\@iiminipage 1
\@imakebox 1
\@imakepicbox 1
\@iminipage 1
\@index 1
\@indexfile ... 207, 1688
\@input 93, 94, 183
\@inputcheck 37
\@ishortstack 1, 1
\@istackcr 1
\@ixstackcr 1
\@keyval 3017, 3020
\@killglue 1
\@largefloatcheck 1
\@latexerr 25, 179, 1641
\@leftmarginskip 378
\@leftmark 1
\@linechar 1
\@linelen 1
\@lowpenalty 1, 76
\@ltxcompatfalse .. 7, 45
\@ltxdoc@PrintChanges 1663, 1675
\@ltxdoc@PrintIndex 1661, 1674
\@ltxdoc@endtheglossary 1666, 1677
\@ltxdoc@theglossary 1665, 1676
\@m 1, 57
\@mainaux .. 93, 94, 183, 205, 1647, 1684
\@mainout 93
\@makebox 1
\@makecaption 369, 2125, 2633
\@makefnmark 1
\@makeother 1
\@makepicbox 1
\@makeschapterhead . 1627
\@marginparreset 1
\@markright 1
\@medpenalty 1
\@minipagerestore 1
\@minus 2550, 2551, 2552, 3077, 3087
\@miv 1
\@mparswitch 378
\@mpfn 1, 382
\@mpfootins 1
\@mplistdepth 1
\@multicnt 1
\@nameuse 205, 1682
\@ne 1, 57
\@newctr 1
\@nil 248
\@nmbrlist 284
\@nobreakfalse . 182, 222
\@nobreaktrue 218

\@nodocument 96
\@noligs 1, 259
\@nomath 1
\@noskipsecfalse 186
\@noskipsectrue .. 1, 183
\@nthm 1
\@oddfoot 224, 226, 229, 241, 1702, 1710, 1713, 1725
\@onlypreamble 249
\@opargbegintheorem ... 1
\@opcol 435
\@othm 1
\@outputbox 435
\@outputpage 434
\@oval 1
\@ovdx 1
\@ovdy 1
\@ovhorz 1
\@ovri 1
\@ovro 1
\@ovvert 1
\@ovxx 1
\@ovyy 1
\@partaux . 93, 178, 186, 187, 188, 204, 1640, 1658, 1659, 1660, 1680
\@partlist . 93, 94, 185, 1652
\@partout 93
\@partsw 93, 94
\@phddoc@PrintChanges 191, 200
\@phddoc@PrintIndex 189, 199
\@phddoc@endtheglossary 194, 202
\@phddoc@theglossary 193, 201
\@picbox 1
\@picht 1
\@picture 1
\@picture@warn 1
\@plus 1011, 1012, 2550, 2551, 2552, 3077, 3087
\@providesfile 1
\@ptsize 1
\@put 1
\@restonecoltrue .. 1600, 2750
\@rightmark 1
\@rightskip 1
\@roman 309, 323
\@savemarbox 1
\@seccntformat ... 1, 65, 244
\@secpenalty 1, 23
\@sect 1
\@sectionspaceoutfalse 511
\@sectionspaceouttrue 510
\@setfps 1
\@setminipage 1
\@setnobreak 1
\@setpar 284
\@sline 1
\@smallverbatim ... 3114, 3132
\@ssect 1, 37, 131
\@stackcr 1
\@startcolumn 435
\@startpbox 1
\@startsection 1, 48, 412, 440, 454, 465, 485, 493, 512, 519
\@starttoc 1
\@strip@args 1
\@svector 1
\@sverb 1
\@svsec . 57, 57, 63, 198
\@svsechd 157
\@sxverbatim 1
\@tempa 185
\@tempb 184, 185
\@tempboxa 1
\@tempboxb 100, 106
\@tempcnta 1
\@tempcntb 1
\@tempdima 1
\@tempdimb 1
\@tempdimc 1
\@tempskipa . 1, 81, 132, 133
\@tempskipb 1
\@tempswafalse 184
\@tempswatrue .. 184, 185
\@temptokena 1
\@test... 440
\@testdef 1
\@text@composite 1
\@text@composite@x 1
\@textsuperscript 1
\@tf@r 1
\@thanks 1
\@thefnmark 382
\@themark 1
\@thispagestyle 434

\@thm 1
\@thmcounter 1
\@thmcountersep 1
\@title 1
\@toplist 435
\@topnum 432
\@toproom 432
\@tryfcolumn 434
\@typeset@till@return 2887, 2888
\@undefined ... 781, 2860
\@unknownversion ... 517, 518
\@upline 1
\@upvector 1
\@verb 1
\@verbatim 1
\@vline 1
\@vobeyspaces 1
\@vvector 1
\@whilenoop 60
\@wholewidth 1
\@wrglossary 1
\@wrindex 1
\@writeckpt 93, 204, 1679
\@writefile 1
\@xarg 1
\@xcentercr 1
\@xdblfloat 1
\@xdim 1
\@xfloat 1
\@xfootnote 1
\@xfootnotenext 1
\@xmpar 1
\@xnthm 1
\@xobeysp ... 1, 88, 580, 825
\@xsect 1
\@xsmallverbatim ... 3132
\@xthm 1
\@xverbatim 1
\@xxxii 1
\@xympar 1, 379
\@yarg 1
\@ympar 1
\@ynthm 1
\@ythm 1
\@yyarg 1
\normalsize 487
\setfontsize 487
\sg 495
\@afterheading 577
\@car 39
\@carcube 39
\@cdr 39
\@cons 38
\@height 34
\@mparswitchfalse ... 634
\BaseDirectory 494
\alphabet 197
\begin 635
\cmd 803
\foo 803
\generate 495
\index 389, 390
\multiput 330
\mylist 38
\newif 46
\normalsize 487
\put 330
\string 45
\bovedisplayshortskip 487
\bovedisplayskip 487
\aselineskip 487
\elowdisplayshortskip 487
\ewboolean 486
\ormalsize 487
\ize 487
\mall 487
\sepackage 486
_ 847
\alloc@ 1
\bm@b 1
\bm@c 1
\bm@l 1
\bm@r 1
\bm@s 1
\bm@t 1
\bottomrule 830, 836
\c@bottomnumber 368
\c@chapter 781
\c@CodelineNo .. 85, 209, 1306, 1308, 1691
\c@dbltopnumber 368
\c@HD@hypercount .. 1006, 1513, 2394, 2484, 2810, 2830
\c@IndexColumns ... 1597, 2748, 2749
\c@page 280
\c@secnumdepth ... 1, 55, 329, 354
\c@section 790
\c@tocdepth 1
\c@topnumber 368

\c@totalnumber 368
\calculate@math@sizes . 1
\check@checksum 1605
\circle 342
\cmd@to@cs 153, 154, 1889, 1902, 3054, 3056
\code@catcode@hook 2900, 2903, 2923
\code@end ... 2945, 2956, 2970, 2993
\code@executefalse . 2895
\code@executetrue .. 2910
\code@flushend 2942, 2953, 2963
\code@flushstart .. 2942, 2952, 2963
\code@mid ... 2943, 2954, 2968, 2972
\code@post .. 2897, 2898, 2902, 2937
\code@pre ... 2896, 2898, 2901, 2937
\code@start . 2941, 2947, 2958
\code@temp ... 2934, 2937
\code@width . 2944, 2955, 2969, 2974
\codeline@wrindex .. 206, 1686, 2375, 2379
\color@begingroup 1
\color@endbox 1
\color@endgroup 1
\color@hbox 1
\color@setgroup 1
\color@vbox 1
\count@ 1
\cp@FOO 93
\curr@math@size 1
\DeclareFontShape@ 1
\default@ds 518
\define@key . 2891, 2892, 2893, 2894, 2895, 2896, 2897, 2898, 2899, 2900
\Describe@Option .. 1492, 1494
\dimen@ 1
\dimen@i 1
\dimen@ii 1
\do@noligs 1
\DocInput 851, 851
\document@default@language 1
\docValue@ 156
\docValue@star 156
\DoNotIndex 821
\ds@ 518
\e@alloc 1
\e@alloc@chardef 1
\e@alloc@top 1
\e@ch@ck 1
\end@dblfloat 1
\endtrivlist 844
\expanded@notin 411
\ext@figure␣=␣'lof' . 369
\ext@TYPE 369
\extract@alph@from@version 1
\f@encoding 1
\f@family 561
\filec@ntents 522
\find@example 2925
\first@arg .. 1759, 1760, 1769, 1770, 1791, 1910, 1912, 2661, 2662
\fnum@ 369
\fnum@table . 2125, 2619, 2633
\fnum@TYPE 369
\fps@ 369
\fps@dbl 1
\frozen@everydisplay .. 1
\frozen@everymath 1
\ftype@ 369
\g@addto@macro 223, 2290
\G@refundefinedfalse 248
\HD@savedestfalse .. 1004
\HD@target .. 1005, 1234, 1509, 2382, 2481, 2807, 2827
\HDorg@encapchar .. 1510, 1513, 2482, 2484, 2808, 2810, 2828, 2830
\HDorg@theCodelineNo 13, 404
\hmode@bgroup 129
\Hy@MakeCurrentHref 1006
\idx@actual ... 114, 162, 169, 1082, 1266, 1276, 1283, 1290, 1294, 1297, 1308, 1315, 1489, 1495, 1797, 1805, 1842, 1849, 1892, 1895, 1900, 1902, 2516, 2522, 2523, 2767, 2768, 2812, 2832, 2833
idx@actual 1080
\idx@encap 1080
\idx@format .. 1080, 1146

\idx@level 168, 294, 1080, 1282, 1293, 1296, 1314, 1493, 1717, 1722, 1734, 1747, 1749, 1750, 1751, 1815, 1842, 1847, 1899, 2523, 2768, 2833, 3167
\idx@quote 1080
\if@afterindent .. 1, 224
\if@DEBUG 12, 19, 39
\if@filesw 183, 186, 204, 206, 400
\if@ignore 1
\if@ltxcompat 7, 46, 409, 435, 461, 484, 490, 511
\if@mainmatter 1
\if@minipage 1
\if@negarg 1
\if@newlist 3133
\if@nobreak 20, 221
\if@noskipsec . 1, 4, 185
\if@ovb 1
\if@ovhline 1
\if@ovl 1
\if@ovr 1
\if@ovt 1
\if@ovvline 1
\if@partsw 184, 1649
\if@tempswa 1, 186
\if@twoside 280
\ifcode@execute ... 2906, 2928
\ifnot@excluded 409
\ifphd@doc@annotate 1051, 1183
\ifphd@doc@colorize 1050, 1177, 1190
\ifphd@doc@toindex . 159, 1049, 1262, 1273, 1290, 1304, 1488
\ifwelsh@first 2202, 2211
\ifwelsh@vowel 2212, 2222
\include 851
\index@excludelist . 413, 821
\index@level 2530
\index@prologue 223, 1603, 1617, 1626, 1700, 2742, 2752
\init@checksum 1606
\input 851
\insc@unt 1
\it@is@a 866, 2317
\K@no@opt@arg 1996, 2003
\K@opt@arg ... 1990, 2003
\Kcyp@no@opt@arg .. 1965, 1966
\Kcyp@opt@arg 1964, 1966
\Kgr@no@opt@arg ... 2251, 2253
\Kgr@opt@arg . 2250, 2253
\kvtcb@col@command ... 60
\kvtcb@col@counter .. 288
\kvtcb@doc@body@command@after 56
\kvtcb@doc@body@command@before 55
\kvtcb@index@command 112
\l@nohyphenation 634
\LT@array 2120
\lt@indexed . 2126, 2141, 2142
\LT@start 2121
\ltx@sh@ft 129
\m@ne 57
\m@th 1, 91, 91
\macro@namepart 412, 821
\math@bgroup 1
\math@egroup 1
\meta 826
\meta@font@select .. 630, 1031, 1039
\meta@hyphen@restore 631, 636
\midrule 836
\N@no@opt@arg 2266, 2275
\N@opt@arg ... 2259, 2276
\nfss@text 1, 628
\Np@no@opt@args ... 2292, 2293
\Np@one@opt@arg ... 2289, 2291
\Np@opt@args . 2289, 2293
\Np@two@opt@args .. 2289, 2290
\ORDINALS@cx 396
\ordinals@cx 389
\origLT@array 2120
\origLT@start 2121
\outer@nobreak 1
\oval 342
\p@cnt 247
\p@enum 298
\part 866, 872
\patch@level 1

\phd@doc@head 757
\phd@doc@index@doc . 1633
\phd@doc@index@off . 1642
\phd@doc@index@pgf . 1632
\phd@doc@index@pgf@ 1595, 1623, 1629
\phd@doc@index@pgfchapter 1625
\phd@doc@index@pgfsection 1616, 1632
\phd@doc@orig@meta 1026, 1032, 1040
\phd@doc@toindexfalse ... 1394, 1420, 1447, 1459
\phd@key@path 1453, 1455, 1457
\phd@layer@pushup .. 1171
\phd@print@LstStyle 287, 292
\phd@Print@Val 176, 177, 1246
\phd@text@lststyles 294, 295, 1069
\phdkv@afterexample .. 80
\phdkv@beforeexample . 79
\phdkv@col@opt . 74, 1008
\phdkv@col@value 67, 1238, 1243, 1248
\phdkv@colhyper . 75, 76, 1148, 1149, 1150, 1151, 1152, 1591, 1592
\phdkv@doc@description 1123, 1352, 1355
\phdkv@doc@indentleft 1115, 1365
\phdkv@doc@indentright 1116, 1366
\phdkv@doc@left ... 1113, 1333
\phdkv@doc@right .. 1114, 1337
\phdkv@doclistingoptions 1075
\phdkv@docmintoptions 1078
\phdkv@docmintstyle 1077
\phdkv@listingencoding 1022
\phdkv@text@color . 1057, 1253
\phdkv@text@colors 1058, 1314
\phdkv@text@env ... 1060, 1213
\phdkv@text@envcontent 1059, 1410
\phdkv@text@envs .. 1061, 1281
\phdkv@text@index . 1064, 1619, 1620, 1627, 1628
\phdkv@text@key ... 1062, 1229
\phdkv@text@keys .. 1063, 1292, 1492
\phdkv@text@pageshort 1055, 1065, 1540
\phdkv@text@value . 1066, 1238
\phdkv@text@values . 167, 1067
\pictur@ 1
\pkdkv@colmeta 77
\protected@edef . 51, 63, 1384, 1409, 1432, 1438
\protected@xdef 358
\Q@no@opt@arg 2302, 2311
\Q@opt@arg 2295
\r@foo 248
\R@no@opt@arg 2382, 2391
\R@opt@arg ... 2376, 2392
\refKey@ 1581, 1587
\refKey@star . 1582, 1586
\rem@pt 1
\saved@indexname ... 843, 1310
\saved@macroname ... 843, 1290
\second@arg . 1911, 1913, 1919
\set@color 1
\setfont@cx 298, 397
\setfontparam@cx ... 303, 402, 427, 448, 476, 504, 526
\sh@ft 129
\sixt@@n 1
\sixt@@n␣ 57
\skip@ 1
\smallverbatim@font 3128, 3134
\strip@pt 1
\tcb@cs 48, 51
\tcb@doc@bfseries ... 288
\tcb@doc@do@description 54
\tcb@doc@toindexfalse 58
\tcb@index@Com .. 49, 124
\tcb@Print@Com ... 47, 60

\tcb@scantokens 288
\tcb@temp@grow@left 1365, 1367
\tcb@temp@grow@right 1366, 1368
\tcb@w@upper@real .. 1371
\telugu@month 2022, 2036
\temp@cx 3013, 3021, 3022, 3024
\texttt 846, 847
\thr@@ 1
\toks@ 1
\toprule 836
\trivlist 844
\tw@ 1, 24, 57
\verb 867
\verb@balance@group ... 1
\verb@egroup 1
\verb@eol@error 1
\verbatim@ ... 2135, 2642
\verbatim@addtoline 2132, 2639
\verbatim@finish .. 2134, 2641
\verbatim@font 1, 20, 259, 567, 825, 867, 2390
\verbatim@nolig@list .. 1
\verbatim@processline 2133, 2640
\verbatim@startline 2131, 2638
\voidb@x 1
\W@no@opt@arg 2429, 2433
\W@opt@arg ... 2425, 2433
\welsh@article 2204, 2265
\welsh@article@do . 2207, 2210
\welsh@firstfalse .. 2218
\welsh@firsttrue ... 2205
\welsh@isvowel 2211, 2223
\welsh@ordinal@long 2232, 2260, 2265
\welsh@ordinalplusxx@long 2255, 2259
\welsh@vowelfalse .. 2214
\welsh@voweltrue .. 2224, 2227, 2229
\WORDS@cx 382
\Words@cx 375
\words@cx 368
\X@no@opt@arg 2441, 2446, 2451, 2452
\X@opt@arg ... 2450, 2452
\xmacro@code . 853, 1727, 2070
\z@ 1, 57
\z@skip 1, 560, 564, 3116, 3117
tex commands:
\tex_clubpenalty ... 188, 200
\tex_day 880
\tex_everypar . 184, 201, 219, 233
\tex_ignorespaces ... 92, 205
\tex_interactionmode 2282, 2285
\tex_lowercase 2067
\tex_month 880
\tex_par 134, 339
\tex_unskip 196
\tex_year 880
\texcolor 1840
texexample environment . 741, 744
\texexample 962
texnote (environment) .. 503, 805
\text 2417
\textbackslash 48, 437, 3060
\textbaht() a765
\textbf 507, 1586, 3175
\textcent(¢) a765
\textcolonmonetary(₡) .. a765
\textcolor 60, 135, 156, 288, 340, 405, 1007, 1008, 1178, 1191, 1207, 1224, 1248, 1483, 1591, 1592, 1911, 1913, 1935, 2534, 3058, 3101, 3103
textcomp (package) 765
\textcurrency(¤) a765
\textcypr 1964, 1965
\textdir ... 1398, 1400, 1415
\textdollar($) a765
\textdong(₫) a765
\texteuro(€) a765
\textfloatsep 481
\textfloatsep 435
\textflorin(ƒ) a765
\textfraction 481
\textfraction 368
\textguarani(₲) a765

\textindent 24
\textit ... 1375, 1385, 2154, 2158, 2715, 2716, 3154
\textlira(£) a765
\textls 297
\textnaira(₦) a765
\textoken 2711
\TextOrMath 150
\textpeso(₱) a765
\textquotesingle 676
\textrm 241
\textrm 1050, 1051, 1052
\textsc 311, 312
\textsf 499
\textsf 455, 456, 1378, 1388, 2505, 2506, 2507
\textsl 297, 2888
\textsterling(£) a765
\textsuperscript 1, 631, 1537
\TextSymbolUnavailable 112
\texttt . 157, 448, 450, 451, 453, 518, 520, 527, 528, 533, 534, 1024, 1179, 1181, 1191, 1192, 1446, 1534, 1550, 1721, 1748, 1757, 1761, 1763, 1771, 1773, 2361, 2474, 2475, 2499, 2501, 2510, 2511, 2515, 2521, 2522, 2523, 2544, 2550, 2551, 2552, 2663, 2666, 2714, 2720, 2721, 2724, 2725, 2726, 2737, 2738, 2767, 2768, 2769, 2781, 2782, 2783, 2784, 2785, 2797, 2798, 2799, 2812, 2815, 2816, 2832, 2833, 2845, 2846, 2847, 2851, 2852, 2853, 3103, 3181, 3190, 3195, 3198, 3201, 3204, 3208, 3212, 3218, 3223, 3227, 3231, 3236
\textunderscore 562
\textvcenter 2556
\textwidth ... 148, 212, 224, 682, 956, 1108, 1704, 2565, 2948
\textwon() a765
\textyen(¥) a765
teXX environment 731
teXXX environment 732
\thanks 499
\thanks 357
\thanks 1, 358
\the 499
thebibliography environment 589
thebibliography (environment) 403
\thechapter 82
\theCNT 247
\theCodelineNo 12, 403
thecs a726
\theenum 298
\thefigure 369
\thefootnote 1
\theglossary 193, 201, 1665, 1676
\theindex .. 1598, 2290, 2750
\themeauthor 2716
themeta a746
\thempfn 1, 382
\thempfootnote 1
theoption a797
\thepage 239, 1535, 1723
\theparagraph 259, 317
\thepart 197, 220, 1671, 1697, 1698, 1716, 1719
thepart commands:
\thepart: 231, 234
\thesection 144, 256
\thesubparagraph 260
\thesubsection 184, 257
\thesubsubsection .. 226, 258
\thetable 2619
thetable commands:
\thetable: ... 2124, 2632
\thetexexp 790, 866, 992
\thetotalsymbols 1933
\thicklines 1, 329
\thickmuskip 231
\thinlines 1, 329
\thinmuskip 231
\thinspace 89
\thinspace 1242, 1243, 1315, 3021, 3025
\thirdlist 111, 115
\thispagestyle 1, 434
\tikz 2848, 2853
\tikzi .. 10, 20, 21, 33, 40, 53, 201, 227, 235, 282, 313, 322, 331, 339, 352, 361, 370, 378
\tiny 1, 362, 405, 662, 2351
\title 357
\title 1

tl commands:
\c_empty_tl 871
\c_space_tl . 1400, 2390, 2427
\tl_clear 698, 1225, 2176, 2177, 2178, 2231, 2238, 2444
\tl_const 61, 63
\tl_count .. 91, 112, 113
\tl_count:n 88
\tl_gclear ... 1617, 1694
\tl_gput_right .. 62, 64, 1700, 2315, 2537
\tl_greplace_all 92, 748
\tl_greplace_all:Nnn . 88
\tl_gset 72, 1605, 2087, 2101, 2123, 2148
\tl_gset_eq 747
\tl_if_empty ... 93, 136, 331, 332, 333, 335, 336, 337, 393, 791, 1089, 1095, 1245, 2222, 2280, 2349, 2385, 2502
\tl_if_empty_p 1086, 1087
\tl_if_eq 1361
\tl_if_head_eq_charcode . . 94, 95, 96, 312, 713, 2449, 2477, 2482, 2491
\tl_if_head_eq_charcode:nNTF 88
\tl_if_head_eq_meaning 97
\tl_if_head_eq_meaning:nNTF 88
\tl_if_in 98, 99, 100, 101, 126, 275, 277, 279, 284, 731, 2081, 2084, 2120, 2445, 2452, 2455, 2458
\tl_if_in:NnTF 88
\tl_if_in:nnTF 88
\tl_map_function 741
\tl_map_inline 610, 803, 2408, 2433
\tl_map_inline:Nn ... 830
\tl_new 15, 16, 38, 40, 47, 48, 51, 56, 57, 58, 60, 60, 68, 69, 69, 70, 78, 82, 83, 84, 1614, 1733, 1810, 2165, 2166, 2167, 2342, 2343
\tl_put_left 673
\tl_put_right . 281, 286, 733, 1405, 1797, 1811, 2149, 2158, 2183, 2188, 2197, 2252, 2270
\tl_remove_all 102, 672, 685, 687, 688, 689
\tl_remove_all:Nn 88
\tl_replace_all 103, 144, 145, 146, 147, 148, 149, 150, 151, 153, 154, 155, 156, 157, 158, 159, 160, 162, 174, 556, 558, 580, 615, 671, 1340, 1444, 2402, 2411, 2425, 2435
\tl_replace_all:Nnn .. 88
\tl_replace_once 12, 104, 287, 403
\tl_replace_once:Nnn . 88
\tl_set ... 39, 121, 250, 251, 252, 253, 254, 267, 270, 274, 549, 550, 552, 570, 588, 595, 609, 670, 678, 868, 873, 893, 933, 943, 1202, 1224, 1241, 1290, 1310, 1339, 1406, 1443, 1697, 1698, 1794, 2143, 2152, 2371, 2399, 2417, 2424, 2442, 2459, 2465, 2472, 2496, 2497, 2506
\tl_set_eq 173, 184, 2333, 2485, 2519
\tl_set_rescan . 555, 613
\tl_tail 2466
\tl_to_str 48, 105, 113, 127, 128, 189, 190, 191, 237, 249, 274, 275, 277, 279, 281, 284, 287, 360, 361, 412, 413, 437, 439, 570, 596, 670, 678, 687, 693, 794, 800, 883, 1020, 1076, 1294, 1330, 1708, 2399, 2402, 2409, 2436, 2443, 2445
\tl_to_str:n ... 88, 814, 817
\tl_use 1613, 1846, 2095, 2109, 2112, 2131, 2159
\tlcolor 1816

\Tm 2413
\Tmesonminus 2186
\Tmesonnull 2187
\Tmesonplus 2188
\Tmoon 2414
\tn . 420, 444, 803, 812, 825
\today 591
\today 1, 7
token commands:
\token_to_str . 144, 145, 146, 147, 153, 154, 155, 156, 162, 174, 187, 260, 297, 421, 435, 1063, 1249, 1445, 2334, 2336, 2337, 2338, 2340, 2390, 2405, 2406, 2409, 2414, 2419, 2427, 2452, 2455, 2458, 2486, 2487, 2505, 2520
\tokencolor 1830
\topfigrule 483
\topfraction 480
\topfraction 368
\topinsert 423
\TOPLIST 435
\topmark 420
\toprule 1116, 2349
\topsep 274, 282
\topskip 287
\topskip 1, 96
totalsymbols counter 770
\tracingfonts 1
\tracingnone 1
\triple 2286
\triplecovbond 2189
\trivlist 293
\tt 259
\ttbf 60
\ttdefault 561
\ttfamily 14, 16, 18, 53, 135, 160, 162, 164, 196, 197, 211, 231, 232, 234, 237, 266, 352, 358, 360, 361, 399, 401, 660, 733, 792, 872, 965, 1003, 1019, 1038, 1041, 1046, 1113, 1207, 1224, 1248, 1325, 1385, 1409, 1433, 1440, 1483, 1517, 1522, 1591, 1669, 1671, 1708, 1716, 1719, 1834, 1962, 1964, 1965, 1971, 1976, 1993, 1999, 2010, 2074, 2082, 2200, 2238, 2246, 2249, 2250, 2251, 2254, 2255, 2257, 2263, 2269, 2282, 2288, 2290, 2291, 2292, 2294, 2299, 2305, 2332, 2345, 2348, 2350, 2352, 2354, 2358, 2363, 2371, 2380, 2386, 2402, 2410, 2413, 2414, 2419, 2421, 2422, 2423, 2424, 2427, 2431, 2434, 2435, 2438, 2440, 2441, 2450, 2451, 2454, 2461, 2487, 2526, 2736, 2814, 2979, 3038, 3044, 3134, 3175
\tvcbox 2555, 2557, 2558
\two@digits 521
\twocolumn 445
\twocolumn[] 96
\twosymbolmessage 2572
\typein 37
typein 1
\typeout 34
\typesetcomment . 2887, 2900, 2980
typography rules
paragraph
indentation 594

U

\uline 1703
\Umd( ) a779
uncial (package) 201
\underbar 1
underline 772
Unicode 203
uninstall 1
\unitlength 328
\unitlength 1, 328, 332, 342
\UnitTested 806
\UnitTested 806, 1392
\unpenalty 2990, 3130
\unrestored@protected@xdef 51
\unsetattribute 542
\unskip 284
\uparrow 2228
\updownarrow 2229
\upshape 1031
\Upsilonmeson 2190
\url 2536
use commands:
\use:n . 315, 1335, 2319, 2346

\use_i:nnn 241
\use_ii:nnn 243
\use_none:n ... 357, 1699
\useasboundingbox 2771, 2775, 2787, 2791, 2801
\usebox 308
\usebox 2558
\usecounter 298
\usecounter 2548
\usefont 165
\usefont 214, 2097
\usefontcmdmessage 2607
\usepackage 515
\usepackage 493, 1926, 2030, 2047, 2049, 2099, 2216, 2322
\UseTextAccent 117
\usetextmathmessage 2597
\UseTextSymbol 117
\UseTextSymbol 109
\utfviii 783
\utfviii 2464
\utfviii 2465

V

\value 147
Values
asdf 727
auto 886
BAR 602
border-top-width 886
case 886
FOO 602
format 886
itemize 299
openbib 601
palette 886
pgf 763
\vardownarrow 2224
\varhookleftarrow 2225
\varhookrightarrow 2226
variable (environment) . 480, 805, 806
variable-sized symbols .. 781
\varleftarrow 2227
\varleftrightarrow 2228
\varlonghookrightarrow . 2229
\varlongleftarrow 2230
\varlongleftrightarrow . 2224
\varlongmapsfrom 2225
\varlongmapsto 2226
\varlongrightarrow 2227
\varmapsfrom 2228
\varmapsto 2229
\varnearrow 2230
\varnwarrow 2224
\varparsep 3074, 3076, 3082, 3086, 3092
\varphoton 2191
\varrightarrow 2225
\varsearrow 2226
\varswarrow 2227
\varuparrow 2228
\varupdownarrow 2229
\vbox 434, 2898
vbox commands:
\vbox_set 1228, 1315
\vbox_unpack_clear 1265, 1317
\vboxcolor 1837
vcoffin commands:
\vcoffin_set:Nnw 956
\vcoffin_set_end: ... 960
\vector 1
\verb 1
\Verbatim 407
\verbatim 259
\verbatim 406
\verbatimchar ... 2291, 2404, 2406, 2409, 2413, 2414, 2419, 2420
\verbatimfamily 20, 48, 311, 313
\verbatimfont . 16, 338, 393, 859, 940, 985, 3021, 3024, 3058, 3060, 3062, 3101, 3103
verse environment 580
version commands:
\l_phd_version 6
\vglue 22
\VIET 1928
\VIETfalse 2368
\Vl 2423
\Vp 782
\Vp 2421
\vphantom 264
\vphantom 188
\Vpl 2424
\vref 2600
\vskip 77, 84
\vspace 78, 86
\vspace 77

W

\water 2192
watermark (package) 425
\Wboson 2193

\Wbosonminus 2194
\Wbosonplus 2195
\wd 106, 2940
\Wf 2434
\whatsit 541
\wide accents 801
Wilson, Peter 207
\wlog 1, 2, 3, 4, 493, 2563, 2564, 2565
\write ... 94, 183, 188, 207, 423
\Wul 2436
\Ww 2435

X

\X 782
\x@protect 51
\xdef 197
XƎLaTeX 191
xindy 398
xkeyval (package) 486
xsect commands:
\xsect:n 94, 96, 100, 105, 108, 110, 159, 163, 172
\xspace 1748, 2465
\Xtemp 3147, 3154

Y

\Y 783
\year ... 147, 175, 219, 263, 330, 375, 418, 443, 502, 543, 589, 634, 678, 721, 765, 808, 851, 894, 1045, 1088, 1131, 1159, 1188, 1218, 1242, 1272, 1315, 1341, 1388, 1417, 1474, 1505, 1549, 1579, 1608, 1638, 1666, 1703, 1732, 1759, 1786, 1814, 1841, 1932, 1960, 1984, 2036, 2064, 2092, 2129, 2170, 2199, 2280, 2288, 2340
\Ytemp 3148, 3154
\yxdim 1

Z

\zap@space 516
\Zboson 2196